

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение
высшего образования**
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
НЕВИННОМЫССКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)»**

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине
«ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ»

Методические указания к выполнению лабораторных работ
для студентов направления подготовки
15.03.04 «Автоматизация технологических процессов и производств»

Невинномысск 2026

Методические указания предназначены для студентов направления подготовки 15.03.04 «Автоматизация технологических процессов и производств». Они содержат основы теории, порядок проведения лабораторных работ и обработки экспериментальных данных, перечень контрольных вопросов для самоподготовки и список рекомендуемой литературы. Работы подобраны и расположены в соответствии с методикой изучения дисциплины «Интеллектуализация систем управления». Объем и последовательность выполнения работ определяются преподавателем в зависимости от количества часов, предусмотренных учебным планом дисциплины, как для очной, так и для заочной форм обучения.

Методические указания разработаны в соответствии с требованиями федерального государственного образовательного стандарта в части содержания и уровня подготовки выпускников по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств»

Составитель: канд. техн. наук А.А. Евдокимов

Ответственный редактор: канд. техн. наук Д.В. Болдырев

Тема 2. Инженерия знаний

Лабораторная работа № 1 «Исследование мозжечковой модели суставного регулятора»

Цель работы

Изучить свойства и возможности ассоциативной нейронной сети СМАС.

Программа работы

1. Исследовать аппроксимирующие возможности сети СМАС.
2. Исследовать обобщающую способность нейронной сети СМАС.
3. Составить и защитить отчет по результатам исследований, в котором должны быть приведены архитектура СМАС с пояснением принципа ее работы, графики изменения ошибки обучения сети СМАС, выводы по обобщающей способности и емкости сети СМАС.

Краткие сведения из теории

СМАС – сокращенное название нейронной сети, которое происходит от первых букв ее полного английского названия: Cerebellar Model Articulation Controller (мозжечковая модель суставного регулятора). Эта нейронная сеть, в основу которой положена нейрофизиологическая модель мозжечка, была разработана американским ученым Альбусом. Первоначально нейронная сеть СМАС была предназначена для управления роботом-манипулятором. В последующих работах Толле и его соавторов было показано, что эта нейронная сеть может быть успешно применена для идентификации и управления нелинейными динамическими объектами.

СМАС предназначена для запоминания, восстановления и интерполяции функций многих переменных. Как и в любой нейронной сети, в СМАС существуют два основных принципиально различных процесса:

- процесс обучения, который осуществляется по измерениям значений функции и ее векторного аргумента с помощью соответствующего алгоритма;
- процесс восстановления, когда по входному вектору восстанавливается или оценивается значение этой функции.

Структура нейронной сети СМАС

Следующие два отличительных момента характеризуют эту нейронную сеть, которая, как отмечалось выше, предназначена для запоминания и восстановления функций многих переменных:

- значения аргументов функции принимают только дискретные значения;
- нелинейное преобразование аргументов функции осуществляется неявно с помощью алгоритма вычисления адресов ячеек ассоциативной памяти, в которых хранятся числа, определяющие значение функции.

Область определения функции в СМАС

Чтобы запомнить функцию многих переменных с помощью нейронной сети СМАС, сначала задается область определения этой функции на гиперпараллелепипеде, вершинами которого являются минимальные и максимальные значения аргументов этой функции; далее ребра гиперпараллелепипеда квантуются с постоянным шагом по каждому ребру, и дискретизованным значениям аргументов присваиваются целочисленные номера. Эти номера однозначно связаны со значениями аргументов функции. Так, для функции одной переменной $y(x)$ аргумент x_i связан со своим номером i следующим соотношением

$$x_i = x_{\min} + (i - 1)\Delta_x, \quad (1)$$

где $\Delta_x = \frac{x_{\max} - x_{\min}}{i_{\max} - 1}$ – шаг квантования; $x_{\min}$, $x_{\max}$ – минимальное и максимальное значения аргумента x из области определения; $i = \overline{1, i_{\max}}$, где $i_{\max}$ – число точек, на которых задана функция $y(x)$.

Номер i можно рассматривать как относительный аргумент функции, поскольку соотношение (1) описывает процедуру сдвига и масштабирования исходной области определения функции. Поэтому в дальнейшем будем считать, что значения аргументов запоминаемой в СМАС функции принимают только целочисленные значения, т. е. функция $y(x)$ от N переменных $x = (x_1, x_2, \dots, x_N)^T$ определена на целочисленной N -мерной сетке $y(x)$:

$$X = \{x_1 = \overline{1, x_{\max}^{(1)}}; x_2 = \overline{1, x_{\max}^{(2)}}; \dots; x_N = \overline{1, x_{\max}^{(N)}}\}. \quad (2)$$

Алгоритм нелинейного преобразования аргументов, или алгоритм вычисления адресов ассоциативной памяти

При построении алгоритма вычисления адресов ассоциативной памяти и алгоритма обучения СМАС ее автор [1] исходил из нейрофизиологического подхода. Согласно этому подходу, каждый входной сигнал возбуждает определенную область мозжечка, суммарная энергия которой соответствует значению запоминаемой функции. В нейронной сети СМАС предполагается, что каждый входной сигнал (аргумент функции) возбуждает, или делает активными, ровно ρ^* ячеек памяти, суммарное содержание которых равно значению запоминаемой функции. Следовательно, даже в случае скалярного аргумента $x = x_1$ ему соответствует ρ^* -мерный вектор номеров активных ячеек памяти. Параметр ρ^* играет очень важную роль. В литературе он известен как обобщающий параметр

(generalization parameter), его значение определяет разрешающую способность СМАС и требуемый объем памяти.

Одномерный случай

Каждому значению скалярного аргумента $x_l = \overline{1, x_{\max}^{(l)}}$ соответствует ровно ρ^* активных ячеек памяти с номерами

$$x_1, x_1 + 1, x_1 + 2, \dots, x_1 + \rho^* - 1, \quad (3)$$

так что максимальному значению аргумента $x_l = x_{\max}^{(l)}$ соответствуют ячейки памяти с номерами $x_{\max}^{(l)}, x_{\max}^{(l)} + 1, x_{\max}^{(l)} + 2, \dots, x_{\max}^{(l)} + \rho^* - 1$, откуда следует, что число ячеек памяти, необходимых для хранения функции одной переменной в СМАС, равно

$$M^{(1)} = x_{\max}^{(1)} + \rho^* - 1, \quad (4)$$

что на $\rho^* - 1$ больше числа значений функции, запоминаемых нейронной сетью.

Таким образом, в одномерном случае нейронная сеть СМАС, с точки зрения требуемого объема памяти, неэффективна, однако ее эффективность проявляется при достаточно точном запоминании гладких функций по небольшому числу обучающей последовательности $n \ll M^{(1)}$.

Очевидно, что номера активных ячеек памяти, соответствующие скалярной переменной $x_l = \overline{1, x_{\max}^{(l)}}$, можно представить ρ^* -мерным вектором $m^{(1)}$, значения компонент которого равны номерам активных ячеек. Из последующего изложения следует, что последовательность значений компонент $m_k^{(l)}$, $k = \overline{1, \rho^*}$, вектора $m^{(1)}$ в одномерном случае не существенна. В многомерном случае она, однако, приобретает принципиальное значение.

Далее приводится алгоритм формирования вектора номеров активных ячеек для одномерного случая, поскольку он служит основой построения вектора номеров активных ячеек в многомерном случае. Этот алгоритм обеспечивает (в соответствии с приведенным в работе [1] требованием) однозначное соответствие номера компоненты k вектора $m^{(1)}$ и ее значения $m_k^{(1)}$.

Введем функцию

$$\overline{a \bmod b} = \begin{cases} b, & a \bmod b = 0; \\ a \bmod b, & a \bmod b \neq 0, \end{cases} \quad (5)$$

где $a \bmod b$ – остаток от деления a на b , где a и b – целые числа.

Тогда значения всех ρ^* компонент $m_k^{(l)}, l = \overline{1, N}, k = \overline{1, \rho^*}$ вектора $m^{(l)}$ вычисляются согласно выражению

$$m_{[(x_i \bmod \rho^* + i) \bmod \rho^*]}^{(l)} = x_i + i, i = \overline{0, \rho^* - 1}. \quad (6)$$

Для вычисления компонент вектора $m^{(l)}$ согласно выражению (6) достаточно определить номер $x_i \bmod \rho^*$ той компоненты, значение которой равно x_i , т. е. вычислить (6) при $i = 0$. Значения следующих компонент равны $x_i + 1$, $x_i + 2$ и т. д., при этом полагается, что номера компонент замкнуты в кольцо, т. е. после компоненты с номером ρ^* следует компонента с номером 1. В качестве поясняющего примера в таблице приведены выражения для вектора $m^{(1)}$ размерности $\rho^* = 8$ для ряда значений аргумента $x_1 = 22, 23, 24, 25, 26, 27, 28, 29, 30$.

Из таблицы следует, что если двум различным значениям аргумента x_1 соответствуют одинаковые номера активных ячеек, то эти номера всегда являются одними и теми же компонентами вектора $m^{(1)}$; при этом первой компонентой вектора $m^{(1)}$ оказываются такие номера ячеек, оста-

ток от деления которых на ρ^* равен единице; второй компонентой вектора $m^{(1)}$ оказываются такие номера ячеек, остаток от деления которых на ρ^* равен двум, и т. д.

Из последовательности (3) и соотношения (6) следует, что если для двух значений аргумента x_1 , равных $x_1 = c$ и $x_1 = d$, выполняется условие

$$|c - d| = \rho < \rho^* \quad (7)$$

$x_1 = 22$	$m^{(1)} = (25, 26, 27, 28, 29, 22, 23, 24) \overline{22 \bmod 8 = 6}$
$x_1 = 23$	$m^{(1)} = (25, 26, 27, 28, 29, 30, 23, 24) \overline{23 \bmod 8 = 7}$
$x_1 = 24$	$m^{(1)} = (25, 26, 27, 28, 29, 30, 31, 24) \overline{24 \bmod 8 = 8}$
$x_1 = 25$	$m^{(1)} = (25, 26, 27, 28, 29, 30, 31, 32) \overline{25 \bmod 8 = 1}$
$x_1 = 26$	$m^{(1)} = (33, 26, 27, 28, 29, 30, 31, 32) \overline{26 \bmod 8 = 2}$
$x_1 = 27$	$m^{(1)} = (33, 34, 27, 28, 29, 30, 31, 32) \overline{27 \bmod 8 = 3}$
$x_1 = 28$	$m^{(1)} = (33, 34, 35, 28, 29, 30, 31, 32) \overline{28 \bmod 8 = 4}$
$x_1 = 29$	$m^{(1)} = (33, 34, 35, 36, 29, 30, 31, 32) \overline{29 \bmod 8 = 5}$
$x_1 = 30$	$m^{(1)} = (33, 34, 35, 36, 37, 30, 31, 32) \overline{30 \bmod 8 = 6}$

то число общих ячеек памяти j , соответствующее этим значениям аргумента, равно $\Delta\rho = \rho^* - \rho$, т. е. чем меньше расстояние между двумя значениями аргумента, тем больше общее число ячеек памяти, соответствующих этим значениям аргумента.

Многомерный случай

В многомерном случае, когда нейронная сеть СМАС запоминает функцию $y(x)$ N переменных $x = (x_1, x_2, \dots, x_N)^T$, заданных на целочислен-

ной сетке $X = \{x_1 = \overline{1, x_{\max}^{(1)}}; x_2 = \overline{1, x_{\max}^{(2)}}; \dots; x_N = \overline{1, x_{\max}^{(N)}}\}$ входному N -мерному вектору x также ставится в соответствие ρ^* -мерный вектор m активных ячеек памяти. Вычисление номеров активных ячеек памяти выполняется с помощью описываемого далее алгоритма.

Каждой компоненте $x_l, l = \overline{1, N}$ вектора x соответствует ρ^* -мерный вектор $m^{(l)}$ активных ячеек памяти. Значения всех ρ^* – компонент вектора $m^{(l)}$ – вычисляются, как это было описано выше в одномерном случае, согласно выражению (6).

В результате этих вычислений вектору x ставится в соответствие промежуточная матрица M активных ячеек памяти с элементами $m_k^{(l)}, l = \overline{1, N}, k = \overline{1, \rho^*}$:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} \rightarrow \begin{pmatrix} m_1^{(1)} & m_2^{(1)} & m_3^{(1)} & \dots & m_{\rho^*}^{(1)} \\ m_1^{(2)} & m_2^{(2)} & m_3^{(2)} & \dots & m_{\rho^*}^{(2)} \\ m_1^{(3)} & m_2^{(3)} & m_3^{(3)} & \dots & m_{\rho^*}^{(3)} \\ \dots & \dots & \dots & \dots & \dots \\ m_1^{(N)} & m_2^{(N)} & m_3^{(N)} & \dots & m_{\rho^*}^{(N)} \end{pmatrix}. \quad (8)$$

На следующем шаге матрице M размера $N \times \rho^*$ ставится в соответствие ρ^* -мерный вектор активных ячеек памяти вектора x . Для этого выполняется последовательное слияние (concatenation) элементов каждого k -го столбца матрицы M в один k -й элемент вектора активных ячеек памяти m , так что этот вектор приобретает следующий вид:

$$m = (m_1^{(1)} m_1^{(2)} m_1^{(3)} \dots m_1^{(N)}; m_2^{(1)} m_2^{(2)} m_2^{(3)} \dots m_2^{(N)}; \dots; m_{\rho^*}^{(1)} m_{\rho^*}^{(2)} m_{\rho^*}^{(3)} \dots m_{\rho^*}^{(N)})^T,$$

каждый элемент которого

$$m_k = m_k^{(1)} m_k^{(2)} m_k^{(3)} \dots m_k^{(N)}, \quad k = \overline{1, \rho^*}, \quad (9)$$

однозначно определяет номер активной ячейки памяти СМАС. Выражение (9) можно трактовать как позиционную запись числа m_k , старший разряд которого равен $m_1^{(1)}$, а младший разряд – $m_1^{(N)}$. К особенностям чисел, образующих символьную запись числа m_k (9), относится отмеченный выше факт, заключающийся в том, что остаток от деления каждого из них на ρ^* равен k . Алгоритм вычисления числа m_k по значениям $m_k^{(l)}$, $l = \overline{1, N}$, принимает наиболее простую форму, когда максимально возможные значения компонент вектора x удовлетворяют следующему соотношению:

$$x_{\max}^{(l)} = (\mu^{(l)} - 1)\rho^* + 1, \quad l = \overline{1, N} \quad (10)$$

Согласно (4), число ячеек памяти, необходимых для хранения функций одной переменной в СМАС, оказывается равным

$$M^{(l)} = \mu^{(l)}\rho^*, \quad l = \overline{1, N}. \quad (11)$$

В этом случае алгоритм вычисления m_k , который приводится без доказательства, имеет вид

$$\begin{aligned} m_k = & (m_k^{(1)} - 1) \prod_{l=2}^N \mu^{(l)} + \text{int}((m_k^{(2)} - 1) / \rho^*) \prod_{l=3}^N \mu^{(l)} + \dots \\ & \dots + \text{int}((m_k^{(N-1)} - 1) / \rho^*) \mu^{(N)} + \text{int}((m_k^{(N)} - 1) / \rho^*) + 1, \quad k = \overline{1, \rho^*}, \end{aligned} \quad (12)$$

где $\text{int}(a/b)$ – функция целочисленного деления a на b .

В качестве иллюстративного примера рассмотрим сеть со следующими параметрами: размерность N входного вектора равна 2, параметр $\rho^* = 4$, $\mu^{(1)} = \mu^{(2)} = 3$, так что согласно (10) $x_{\max}^{(1)} = x_{\max}^{(2)} = 9$ и, следовательно, $X = \{\overline{1, 9}; \overline{1, 9}\}$ (2).

Найдем номера ячеек памяти СМАС согласно выражениям (6), (8), (9) и (12), которые соответствуют входному вектору $x^{(1)} = (6; 8)^T$:

$$x[1] = \begin{vmatrix} 6 \\ 8 \end{vmatrix} \rightarrow M = \begin{vmatrix} 9 & 6 & 7 & 8 \\ 9 & 10 & 11 & 8 \end{vmatrix} \rightarrow m[1] = (9, 9; 6, 10; 7, 11; 8, 8)^T.$$

Пересчитывая символические номера 9,9; 6,10; 7,11; 8,8 активных ячеек в соответствии с алгоритмом (12), получим

$$9,9 \rightarrow 8 * 3 + 2 + 1 = 27,$$

$$6,10 \rightarrow 5 * 3 + 2 + 1 = 18,$$

$$7,11 \rightarrow 6 * 3 + 2 + 1 = 23,$$

$$8,8 \rightarrow 7 * 3 + 1 + 1 = 23,$$

так что вектор номеров активных ячеек, соответствующий вектору $x^{(1)} = (6; 8)^T$, равен

$$m[1] = (27, 18, 21, 23)^T.$$

Не приводя промежуточных результатов, заметим, что вектор $x^{(2)} = (7; 6)^T$ соответствует вектору номеров активных ячеек в символическом $m[2] = (9, 9; 10, 6; 7, 7; 8, 8)^T$ и в десятичном $m[2] = (27, 29, 20, 23)^T$ представлениях, и, следовательно, ячейки памяти с номерами 23 и 27 активируются как вектором, так и вектором $x[2]$.

Отметим здесь, что общее число ячеек памяти для рассматриваемой сети $M = 36$, тогда как возможное число точек запоминаемой функции равно $\overline{M} = 81$.

Объем памяти

Необходимое число ячеек памяти M нейронной сети СМАС следует из формулы (12), когда все числа $m_k^{(l)}, l = \overline{1, N}$, принимают свои максимально возможные значения, т. е. когда $m_k^{(l)} = M^{(l)} = \mu^{(l)} \rho^*, l = \overline{1, N}$.

При этом сомножители $\text{int}((m_k^{(l)} - 1) / \rho^*), l = \overline{1, N}$, входящие в формулу (12), оказываются равными

$$\text{int}((m_k^{(l)} - 1) / \rho^*) = \mu^{(l)} - 1, l = \overline{1, N}, \quad (13)$$

а величина M задается выражением

$$\begin{aligned} M &= (\mu^{(1)} \rho^* - 1) \prod_{l=2}^N \mu^{(l)} + (\mu^{(2)} - 1) \prod_{l=3}^N \mu^{(l)} + \dots \\ &\dots + (\mu^{(N-1)} - 1) \mu^{(N)} + (\mu^{(N)} - 1) + 1 = \rho^* \prod_{l=1}^N \mu^{(l)}. \end{aligned} \quad (14)$$

Соотношения (10) и (14) позволяют выразить объем памяти СМАС в функции $x_{\max}^{(l)}$, $l = \overline{1, N}$, где $x_{\max}^{(l)}$ — наибольшее значение, которое принимает аргумент x_l :

$$\begin{aligned} M &= \rho^* \prod_{l=1}^N \mu^{(l)} = \rho^* \prod_{l=1}^N \frac{(\mu^{(l)} \rho^* - \rho^* + 1) + \rho^* - 1}{\rho^*} = \\ &= \rho^* \prod_{l=1}^N \frac{x_{\max}^{(l)} + \rho^* - 1}{\rho^*} = \rho^* \prod_{l=1}^N \frac{M^{(l)}}{\rho^*}, \end{aligned} \quad (15)$$

где $M^{(l)}$ определено соотношением (11).

Если бы функция многих переменных $y(x)$ хранилась в памяти обычным способом, когда значение функции записывается в одну ячейку памяти, то для этого, очевидно, понадобилось бы

$$\overline{M} = \prod_{l=1}^N x_{\max}^{(l)} \quad (16)$$

ячеек памяти, что следует также из формулы (15) при $\rho^* = 1$, когда в СМАС каждый входной сигнал возбуждает только одну ячейку памяти. Из сравнения выражений (15) и (16) следует, что СМАС уменьшает объем памяти по сравнению с обычным способом хранения приблизительно в $(\rho^*)^{(N-1)}$ раз, так что эффективность его с точки зрения требуемого объема памяти растет с увеличением размерности входного вектора.

Пользуясь выражением (15), можно исследовать влияние параметра ρ^* на объем памяти СМАС при фиксированных характеристиках вход-

ного вектора x . Наиболее удобно сделать это, когда функция $y(x)$ задана на гиперкубе, т. е. когда все значения $x_{\max}^{(l)}, l = \overline{1, N}$, одинаковы и равны

$$x_{\max}^{\circ} = 2^D + 1. \quad (17)$$

Полагая

$$\rho^*(r) = 2^r, r = \overline{0, D}, \quad (18)$$

найдем $\mu^{\circ} = \mu^{(l)}$ (10) и $M^{\circ} = M^{(l)}$ (11):

$$\mu^{\circ} = 2^{D-r} + 1; M^{\circ} = \mu^{\circ} \cdot \rho^*(r) = 2^D + 2^r. \quad (19)$$

Объем памяти СМАС согласно (15) в этом случае равен

$$M = \frac{(2^D + 2^r)^N}{2^{r(N-1)}}. \quad (20)$$

Нетрудно показать, что объем памяти M (20) с ростом параметра ρ^* убывает и достигает минимально возможного значения

$$M_{\min} = \frac{(2^D + 2^D)^N}{2^{D(N-1)}} = 2^N \cdot 2^D. \quad (21)$$

при $\rho^*(r) = 2^D$. Таким образом, коэффициент выигрыша объема памяти η , определяемый как отношение объема памяти $\overline{M}$ (16) при обычном способе хранения информации к объему памяти СМАС $M_{\min}$, принимает наибольшее значение при $\rho^*(r) = 2^D$ и оказывается равным

$$\eta = \frac{\overline{M}}{M_{\min}} = \frac{(2^D + 1)^N}{2^N \cdot 2^D}.$$

При $D \geq 5$ выражение для коэффициента η с большой точностью можно принять равным $\eta = 2^{DN-D-N}$. В частности, если хранить изображение, заданное на двумерной сетке $(2^{10} + 1)(2^{10} + 1)$ обычным способом, то

потребуется один мегабайт памяти, тогда как если это изображение запомнить в СМАС при максимально возможном значении коэффициента $\rho^*(r) = 2^{10}$, то можно ограничиться только четырьмя килобайтами памяти.

Выход нейронной сети СМАС

Значение запомненной в СМАС функции $y(x)$ равно сумме содержимого активных ячеек памяти. Для формализации понятия «выход сети» введем следующие переменные:

$w[n]$ – M -мерный вектор памяти сети, вычисленный на n -м шаге обучения, каждая j -я компонента которого $w_j[n]$, $j = \overline{1, M}$, соответствует содержимому j -й ячейки памяти СМАС;

$a(x)$ – M -мерный ассоциативный вектор, однозначно связанный с вектором аргументов x посредством вектора m (9) номеров активных ячеек памяти по следующему правилу: элементы вектора a_j , $j = \overline{1, M}$, номера которых совпадают с номерами активных ячеек памяти, равны единице, все остальные элементы вектора a равны нулю.

Следовательно, в векторе a всегда ρ^* элементов равны единице. Тогда в соответствии с правилом функционирования нейронной сети СМАС ее выход $y(x, n)$ при заданном входном векторе x равен

$$y(x, n) = a^T(x)w[n], \quad (22)$$

т. е. выход равен сумме содержимого активных ячеек памяти сети. Такое представление выхода сети было введено в работе Милитцера и Паркса [13]; оно позволило понять природу этой нейронной сети, исследовать процесс обучения в ней и найти пути повышения ее эффективности.

Обучение нейронной сети СМАС

Процесс формирования M -мерного вектора памяти w , направленный на запоминание функции $y(x)$ N переменных $x = (x_1, x_2, \dots, x_N)^T$, заданных на целочисленной сетке $X = \{x_1 = \overline{1, x_{\max}^{(1)}}; x_2 = \overline{1, x_{\max}^{(2)}}; \dots; x_N = \overline{1, x_{\max}^{(N)}}\}$ называется *обучением*. Алгоритм обучения нейронной сети СМАС был предложен Альбусом в работе [1]. Этот алгоритм, имеющий рекуррентную природу, функционирует следующим образом.

Пусть после $(n-1)$ -го измерения значений запоминаемой функции $y(x)$ и соответствующих значений вектора аргументов $x(x[i], y[i], i = \overline{1, n-1})$ был вычислен вектор памяти $w[n-1]$. Тогда на следующем n -м шаге после измерения значения функции $y[n]$ при известном значении аргумента $x[n]$ сначала с помощью «алгоритма нелинейного преобразования аргументов» вычисляются номера активных ячеек памяти, т. е. вычисляется соответствующий вектору $x[n]$ вектор $m[k]$ (9, 12), далее вычисляется предсказываемое нейронной сетью значение функции $y[k]$, равное сумме содержимого активных ячеек памяти. Вычисляются ошибка предсказания $\varepsilon[n] = y[n] - \hat{y}[n]$ и величина коррекции $\Delta w[k] = \varepsilon[n] / \rho^*$, которая прибавляется к содержимому активных ячеек памяти. Неактивные ячейки коррекции не подвергаются.

Если воспользоваться ассоциативным вектором a , однозначно связанным с вектором m (9) номеров активных ячеек памяти по следующему правилу: элементы вектора a_j , $j = \overline{1, M}$, номера которых совпадают с номерами активных ячеек памяти, равны единице, все остальные элементы вектора a равны нулю, то алгоритм обучения СМАС можно записать в следующей форме:

$$w_i[n] = w_i[n-1] + (y[n] - \sum_j w_j[n-1]) / \rho^*; \quad (23)$$

$$\forall i a_i[n] = 1; \forall j a_j[n] = 1;$$

$$w_i[n] = w_i[n-1], \forall i a_i[n] = 0. \quad (24)$$

Последовательность $w[n]$ (23), (24) устроена так, что после коррекции вектора памяти $w[n-1]$ оценка функции $y[n]$, которая является суммой содержимого активных ячеек, совпадает с ее значением, т. е.

$$y[n] = \sum_j w_j[n], \forall j a_j[n] = 1 \quad (25)$$

Это утверждение следует непосредственно из суммирования правой и левой частей уравнения (23) по активным номерам ячеек памяти.

Альбус исследовал сходимость алгоритма только экспериментально. Теоретическое обоснование сходимости последовательности (23), (24) впервые было дано в работе Милитцера и Паркса [13, 30], которые показали, что алгоритм обучения СМАС есть итеративный процесс решения системы линейных уравнений относительно неизвестных значений компонент вектора памяти w . Вопросы сходимости процедуры (23), (24) обсуждались также в работе [31]. В работе [13] было показано, что в зависимости от вида запоминаемой функции и параметров СМАС последовательность $w[n]$ (23), (24) сходится либо в точку, либо в некоторую область, которую авторы называли *зоной притяжения*. Основой доказательства сходимости последовательности $w[n]$ послужило представление алгоритма (23), (24) в эквивалентной векторной форме:

$$w[n] = w[n-1] + \frac{y[n] - a^T(x[n])w[n-1]}{a^T(x[n])a(x[n])} a(x[n]), \quad (26)$$

$$w[0] = 0, n = 1, 2, \dots$$

В работе [14] было отмечено, что предложенный Альбусом алгоритм обучения в форме (26) является известным алгоритмом Качмажа [16, 17] для итеративного решения системы линейных уравнений вида

$$a^T(x[n])w = y[n], n = 1, 2, \dots, \quad (27)$$

и поэтому в дальнейшем алгоритм (23), (24) обучения нейронной сети СМАС будем называть *алгоритмом Альбуса—Качмажа*. Представление алгоритма обучения в форме (26) позволило в работе [14] предложить и исследовать алгоритм обучения, обладающий повышенной скоростью сходимости, который будет описан во второй части статьи. С помощью такого представления также удалось повысить точность восстановления запоминаемых функций [32].

Система уравнений (27) обладает рядом особенностей: в каждом уравнении системы ровно ρ^* коэффициентов равны единице, все остальные коэффициенты равны нулю; как правило, размерность M вектора w существенно больше числа ненулевых элементов вектора, число которых равно ρ^* , так что матрица системы (27) является сильно разреженной матрицей; поскольку максимально возможное число линейно независимых уравнений системы (27) равно числу неизвестных, равному размерности M вектора w , а максимальное число уравнений равно числу точек

$$\overline{M} = \prod_{l=1}^N x_{\max}^{(l)} \text{ целочисленной сетки}$$

$$X = \{x_1 = \overline{1, x_{\max}^{(1)}}; x_2 = \overline{1, x_{\max}^{(2)}}; \dots; x_N = \overline{1, x_{\max}^{(N)}}\}$$

и, как правило, $\overline{M} \gg M$ (напомним, что $M = \prod_{l=1}^N (x_{\max}^{(l)} + \rho^* + 1) / (\rho^*)^{N-1}$

(15)), то эта система уравнений обычно несовместна; система уравнений (27) всегда имеет единственное решение в случае, когда параметр $\rho^* = 1$; при этом матрица системы равна единичной матрице или легко преобра-

зуется к ней путем перенумерования последовательности $y[n]$. Этот случай соответствует обычному способу хранения информации, когда в каждой ячейке памяти хранится значение функции, а номер ячейки памяти однозначно связан с аргументом функции.

Несовместность системы уравнений (27) приводит к тому, что последовательность $w[n]$ (26) сходится в зону решения («зону притяжения» [13]) и совершает там движения в соответствии с алгоритмом обучения сети (26) и законом формирования входного вектора $x[n]$. За решение принимается одно из значений вектора $w[n]$, которое принадлежит зоне притяжения.

Последовательность выполнения работы

1. Получить у преподавателя функцию двух переменных
2. Используя файл `smas.mcd` обучить нейронную сеть в MathCAD воспроизводить заданную функцию двух переменных.

Содержание отчета

1. Название лабораторной работы, цель и программа работы.
2. Коды Mathcad, описывающие структуру сети СМАС и ее обучение с пояснениями и промежуточными результатами (для массивов большой размерности все значения изображать необязательно).
3. Графики оригинальной функции в заданной области определения и функции воспроизводимой сетью СМАС.
4. График изменения абсолютной и относительной ошибки обучения сети. Итоговая ошибка обучения СМАС. 5. Выводы об информационной емкости, структуре, объеме памяти по сравнению с классическими методами табличной реализации заданной функции, скоростью обучения, способностью обобщения результатов обучения и других свойств СМАС.

Контрольные вопросы к защите лабораторной работы

1. Пояснить процесс формирования виртуальных и физических адресов сети СМАС.
2. На чем основывается способность нейронной сети обучаться с использованием предыдущего опыта?
3. За счет чего сокращается объем памяти сети СМАС по сравнению с классическим способом хранения функций табличной арифметики?
4. Показать зависимость информационной емкости сети от ее параметров.

Тема 2. Инженерия знаний

Лабораторная работа № 2 «Многослойные нейронные сети»

Цель работы

Изучить метод обучения обратного распространения ошибки многослойных нейронных сетей при использовании сети для различных прикладных задач

Программа работы

1. Исследовать возможности многослойных нейронных сетей при решении задач аппроксимации функции многих переменных.
2. Исследовать возможности многослойных нейронных сетей при решении задач прогноза значений временного ряда
3. Исследовать возможности многослойных нейронных сетей при решении задач классификация векторных данных
4. Составить и защитить отчет по результатам исследований.

Краткие сведения из теории

Архитектура многослойной нейронной сети представлена на рисунке 1. Все нейроны сети объединены в группы, называемые слоями. За каждый такт дискретного времени сигнал передается от слоя к слою, пока не достигнет выходного (последнего) слоя, на котором фиксируется реакция сети. Входной (нулевой) слой не выполняет какой-либо вычислительной функции и служит исключительно для распределения входного векторного сигнала $n = (n_1, n_2, \dots, n_{N_0})$ на нейроны первого слоя. В многослойной нейронной сети сигнал распространяется только в прямом направлении: обратные связи и связи между нейронами одного слоя отсутствуют. В связи с этим представленная на рис. 1 сеть называется мно-

гослойной нейронной сетью прямого распространения (feed-forward layered neural network).

Рабочие слои сети, расположенные между нулевым (распределительным) и выходным слоями, называются скрытыми (hidden layers). Состояние q -го слоя характеризуется вектором s^q , $q = \overline{0, K}$. В соответствии с обозначением входного векторного сигнала справедливо $s^0 = n$. Число нейронов в q -ом слое принимается равным N_q , так что $s^q = (s^q_1, s^q_2, \dots, s^q_{N_q})$, $q = \overline{0, K}$. Здесь s^q_i , $q = \overline{0, K}$, $i = \overline{1, N_q}$ означает состояние i -го нейрона q -го слоя.

Функционирование каждого слоя сети определяется значениями матрицы W синаптических коэффициентов и вектора $-b$ смещений нейронов. В q -ом слое, $q = \overline{1, K}$, матрица W^q имеет размерность $[N_q, N_{q-1}]$, а вектор $b^q = (b^q_1, b^q_2, \dots, b^q_{N_q})$. Расчет прохождения сигнала в сети прямого распространения выполняется по следующим формулам:

$$\begin{cases} s^q_i = f_q(h^q_i), \\ h^q_i = \sum_{j=1}^{N_{q-1}} w^q_{ij} s^{q-1}_j - b^q_i, \end{cases} \quad (2.1)$$

$$q = \overline{1, K}, i = \overline{1, N_q},$$

где h^q_i — потенциал i -го нейрона q -го слоя; $f_q(h)$ — функция активации нейронов q -го слоя (предполагается, что все нейроны одного слоя имеют одинаковые функции активации). К функциям активации $f_q(h)$, $q = \overline{1, K}$, предъявляется требование непрерывности и дифференцируемости.

“Жесткие” пороговые функции активации не допускаются. Под w^q_{ij} в формуле (2.1) понимается элемент матрицы W^q синаптических коэффициентов q -го слоя.

Удобно преобразовать математическую модель (2.1) путем введения одного дополнительного нейрона в каждый слой МНС. Отметим эти

нейроны нулевым индексом и положим их состояние тождественно равным 1, так что $s_0^q = 1$ для $q = \overline{0, K}$. За счет введенных нейронов естественно расширяются вектора s^q , $q = \overline{0, K}$, состояний нейронов в слоях:

$$\bar{s}^q = (s_0^q, s_1^q, s_2^q, \dots, s_{N_1}^q), \quad q = \overline{0, K}. \quad (2.2)$$

Расширим также матрицу синаптических коэффициентов в q -ом слое ($q = \overline{1, K}$), дополнив ее нулевым столбцом, содержащим смещения нейронов q -го слоя:

$$\bar{W}^q = \begin{pmatrix} -b_1^q & w_{11}^q & w_{12}^q & \dots & w_{1N_{q-1}}^q \\ -b_2^q & w_{21}^q & w_{22}^q & \dots & w_{2N_{q-1}}^q \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -b_{N_q}^q & w_{N_q 1}^q & w_{N_q 2}^q & \dots & w_{N_q N_{q-1}}^q \end{pmatrix} \quad (2.3)$$

Обозначим $\bar{w}_{ij}^q$, $i = \overline{1, N_q}$, $j = \overline{0, N_{q-1}}$, матрицы $\bar{W}^q$. В новых обозначениях математическая модель (2.1) преобразуется к следующему виду:

$$\begin{cases} s_0^q = 1, & s_i^q = f_q(h_i^q), \\ h_i^q = \sum_{j=0}^{N_{q-1}} w_{ij}^q s_j^{q-1} - b_i^q, \end{cases} \quad (2.4)$$

$$q = \overline{1, K}, \quad i = \overline{1, N_q}.$$

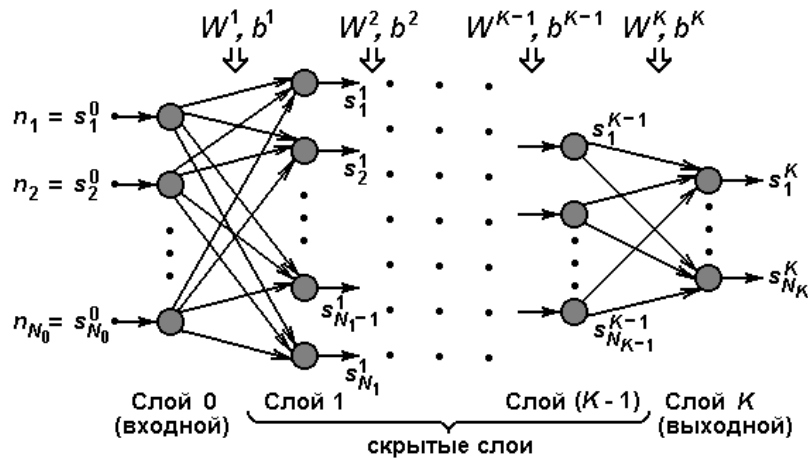


Рисунок 2.1 – Архитектура многослойной нейронной сети

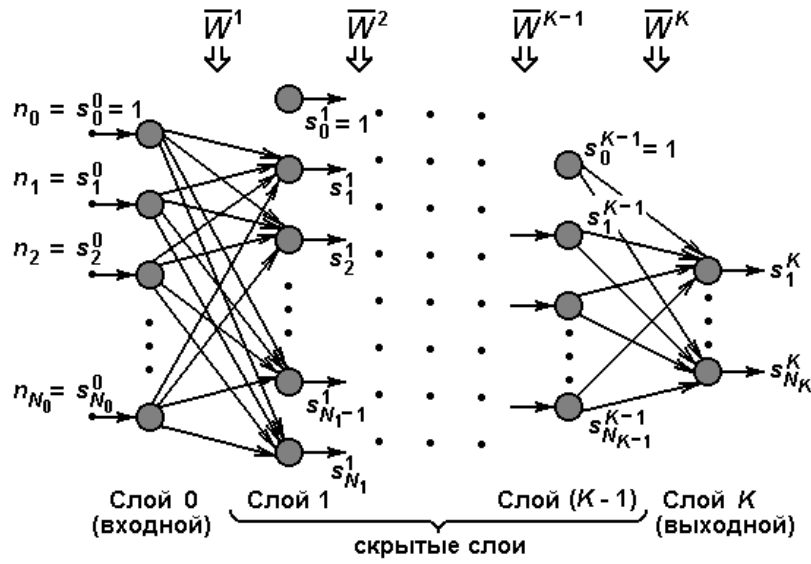


Рисунок 2.2 – Эквивалентное преобразование многослойной нейронной сети

Формулам (2.4) соответствует схема функционирования МНС, представленная на рисунке 2.2.

Применяя векторно-матричные операции над данными в слоях МНС, перейдем от уравнений (2.4) к следующей математической модели:

$$\begin{cases} s_0^q = 1, s_i^q = f_q(h_i^q), \\ h^l = \bar{s}^{q-1}(\bar{W}^q)^T, \end{cases} \quad (2.5)$$

$$q = \overline{1, K}, i = \overline{1, N_q},$$

где $h^q = (h_1^q, h_2^q, \dots, h_{N_q}^q)$ — вектор потенциалов нейронов q -го слоя, верхний индекс T — знак транспонирования вектора или матрицы.

Постановка задачи обучения МНС

Многослойная нейронная сеть осуществляет преобразование входного вектора n размерности $N_0 = M$ в выходной вектор s^K размерности $N_K = M$: $s^K = \varphi(n)$. Это функциональное преобразование для выбранной архитектуры сети (числа слоев, распределения нейронов по слоям и активационных характеристик нейронов) зависит от значений синаптических ко-

эффицентов и смещений всех нейронов. Эти параметры сети собраны в матрицы $\bar{W}^q$, $q = \overline{1, K}$. Даже в достаточно простых практических приложениях число параметров сети может достигать нескольких десятков тысяч. Например, в задаче распознавания изображения, представленного матрицей значений яркости размером 10×10 ($M = 100$), с помощью сети, содержащей два рабочих слоя (скрытый слой $N_1 = 50$ и выходной слой $N_2 = N = 2$), число параметров превышает 5000. Если при решении практической задачи разумно подобрана архитектура нейронной сети, то настройкой параметров можно добиться близости фактического функционального преобразования, реализуемого нейронной сетью, к желаемому преобразованию в решаемой задаче. Процесс настройки параметров нейронной сети называется ее обучением.

Для обучения МНС используются данные обучающей выборки. Выборка состоит из образцов (примеров), содержащих желаемую реакцию $\sigma^p = (\sigma^p_1, \sigma^p_2, \dots, \sigma^p_N)$ сети на входное воздействие $n^p = (n^p_1, n^p_2, \dots, n^p_M)$:

$$\{ n^p, \sigma^p \}, p = \overline{1, P}, \quad (2.6)$$

где P — объем обучающей выборки.

Обозначим фактическую реакцию рассматриваемой МНС на воздействие n^p через s^{pK} (верхний индекс K соответствует номеру выходного слоя). Тогда разность $(\sigma^p - s^{pK})$ характеризует ошибку преобразования входного сигнала сетью, а показатель

$$D = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^N (\sigma_m^p - s_m^{pK})^2 \quad (2.7)$$

может служить критерием качества настройки параметров нейронной сети. Задача обучения нейронной сети сводится в такой постановке к достижению

$$\min_{\bar{W}^q, q=\overline{1, K}} D(\bar{W}^1, \bar{W}^2, \dots, \bar{W}^K). \quad (2.8)$$

Поскольку при такой формализации задачи обучения предполагаются известными желаемые реакции МНС на входные сигналы из заданной выборки, что равносильно присутствию “учителя” в процессе обучения, сам процесс обучения называется “обучением с учителем”.

Градиентный поиск оптимальных параметров МНС

Определим закон, по которому эволюционируют значения параметров $\bar{w}_{ij}^q$, $q = \overline{1, K}$, $i = \overline{1, N_q}$, $j = \overline{0, N_{q-1}}$, в процессе минимизации показателя D , следующим равенством:

$$\frac{d}{dt} \bar{w}_{ij}^q = -\varepsilon \frac{\partial}{\partial \bar{w}_{ij}^q} D \quad (2.9)$$

где ε — параметр закона обучения, время t процедуры настройки полагается непрерывным. Если объединить все настраиваемые параметры МНС в один вектор, то закон (2.9) определяет, что в процессе эволюции этот вектор изменяется в каждый момент времени в направлении антиградиента критерия D . Несложно показать, что закон обучения (2.9) обеспечивает невозрастание показателя D в процессе эволюции:

$$\frac{d}{dt} D = \sum_{q=1}^K \sum_{i=1}^{N_q} \sum_{j=1}^{N_{q-1}} \frac{\partial}{\partial \bar{w}_{ij}^q} D \frac{d}{dt} \bar{w}_{ij}^q = -\varepsilon \sum_{q,i,j} \left(\frac{\partial}{\partial \bar{w}_{ij}^q} D \right)^2 \leq 0.$$

Таким образом, следуя закону (2.9) настройки параметров МНС, из любого начального значения параметров осуществляется спуск к экстремальной точке. В связи с тем, что показатель D , рассматриваемый как функция настраиваемых параметров, может содержать много локальных минимумов, нет гарантии спуска из произвольного начального значения вектора параметров в точку глобального минимума. В практических приложениях рекомендуется реализовать процедуру градиентного спуска неоднократно из различных начальных положений для выбора лучшего из полученных решений.

Численная реализация закона (2.9) требует перехода от непрерывного к дискретному времени $t = 0, 1, 2, \dots$. В этом случае уравнения эволюции синаптических коэффициентов записываются в следующей форме:

$$\bar{w}_{ij}^q(t+1) = \bar{w}_{ij}^q(t) - \alpha \frac{\partial}{\partial \bar{w}_{ij}^q} D(t), \quad (2.10)$$

где α — параметр закона обучения. От выбора параметра α зависит скорость обучения, форма переходного процесса по настраиваемым параметрам, а также сам факт сходимости процесса обучения. Чем меньше значение параметра α , тем менее различаются уравнения (2.9) и (2.10), реализующие поиск в непрерывном и дискретном времени. Следовательно, тем выше вероятность сходимости дискретной поисковой процедуры (гарантированное невозрастание критерия D в процессе эволюции доказано выше только для непрерывного времени). В то же время чрезмерное уменьшение параметра α затягивает переходные процессы, увеличивает необходимое время вычислений. Обычно в программных средствах предусматриваются различные возможности адаптивной подстройки значения α в процессе поиска экстремума, а выбор начального значения этого параметра определяется пользователем.

Для реализации закона обучения (2.10) необходимо определить алгоритм вычисления частных производных критерия D по искомым параметрам:

$\frac{\partial}{\partial \bar{w}_{ij}^q} D$, $q = \overline{1, K}$, $i = \overline{1, N_q}$, $j = \overline{0, N_{q-1}}$. Этому вопросу посвящены следующие

два подраздела.

Алгоритм обучения однослойной нейронной сети с непрерывной передаточной функцией

Рассмотрим частный случай $K = 1$, когда сеть содержит один рабочий слой. Опустим верхний индекс номера слоя в обозначениях настраи-

ваемых синаптических коэффициентов $\bar{w}_{ij}^q$, $i = \overline{1, N}$ ($N = N_1$), $j = \overline{1, M}$ ($M = N_0$), а также векторов потенциалов h_i , $i = \overline{1, N}$, и состояний нейронов s_i , $i = \overline{1, N}$, выходного слоя. Опустим также индекс номера слоя в обозначении передаточной функции нейронов $f(h)$.

В принятых обозначениях критерий точности D на основании выражений (1.7) и (1.4) может быть представлен в следующем виде:

$$D = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^N (\sigma_m^p - s_m^{pk})^2 = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^N (\sigma_m^p - f(h_m^p))^2 \quad (2.11)$$

где $h_m^p = \sum_{j=0}^M \bar{w}_{mj} n_j^p$, ($n_j = s_j^0$, $j = \overline{1, M}$) ; n_j^p — j -я составляющая входного вектора n в p -м примере обучающей выборки.

Вычислим с использованием выражения (2.11) производную показателя D по аргументу $\bar{w}_{ij}$, $i = \overline{1, N}$, $j = \overline{1, M}$:

$$\begin{aligned} \frac{\partial}{\partial \bar{w}_{ij}} D &= - \sum_{p=1}^P (\sigma_i^p - f(h_i^p)) \frac{\partial}{\partial \bar{w}_{ij}} f(h_i^p) = \\ &= - \sum_{p=1}^P (\sigma_i^p - f(h_i^p)) f'(h_i^p) n_j^p \end{aligned} \quad (2.12)$$

В выводе выражения (2.12) было учтено, что в сумме слагаемых по индексу m в выражении (2.11) только одно слагаемое, соответствующее значению $m = i$, зависит от $\bar{w}_{ij}$. В связи с этим окончательное выражение (2.12) содержит лишь сумму по примерам обучающей выборки. Введем обозначение:

$$\begin{aligned} \Delta_i^p &= (\sigma_i^p - f(h_i^p)) f'(h_i^p) = (\sigma_i^p - s_i^p) f'(h_i^p), \\ i &= \overline{1, N}. \end{aligned} \quad (2.13)$$

Тогда выражение (1.12) преобразуется к следующей краткой форме:

$$\frac{\partial}{\partial \bar{w}_{ij}} D = - \sum_{p=1}^P \Delta_i^p n_j^p, \quad i = \overline{1, N}, \quad j = \overline{0, M}. \quad (2.14)$$

Заметим, что в силу предположения о непрерывности и дифференцируемости активационной характеристики нейронов МНС производная $f'(h)$ существует (см. выражения (2.12) и (2.13)). В частности, для $f(h) = \text{th}(\beta h)$

$$f'(h) = \beta [1 - (f(h))^2],$$

а для логистической функции $f(h) = 1 / (1 + \exp(-\beta h))$

$$f'(h) = \beta f(h) (1 - f(h)).$$

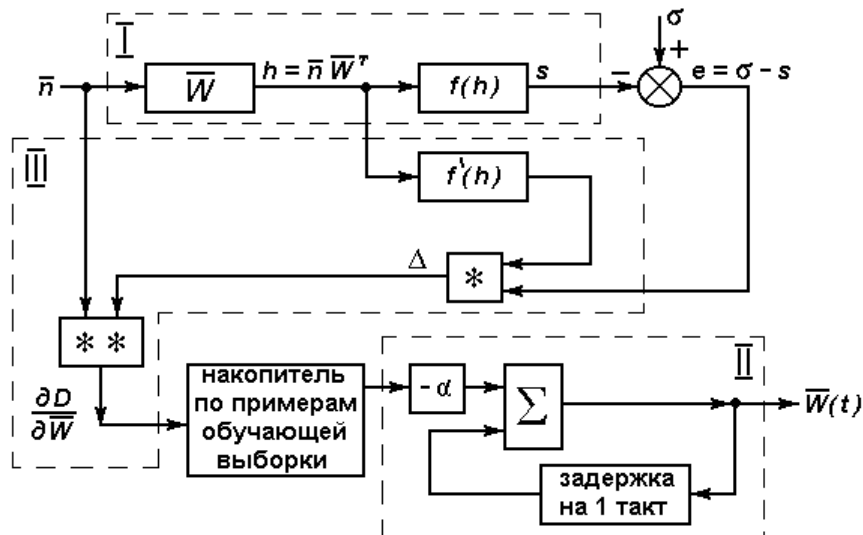


Рисунок 2.3 – Схема настройки параметров однослойной нейронной сети

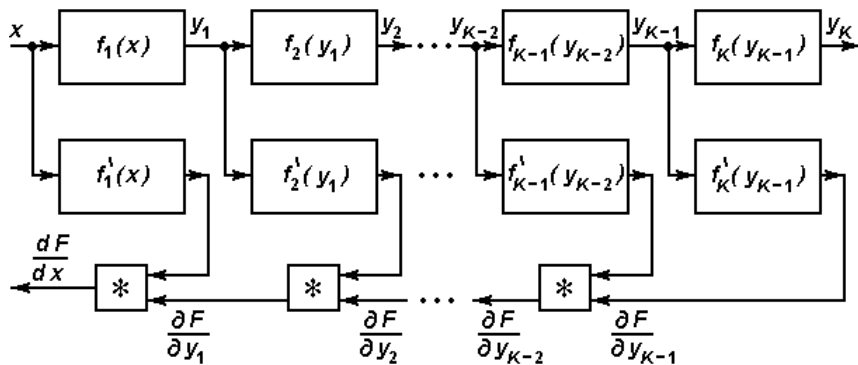


Рисунок 2.4 – Схема вычисления производной при последовательности соединений нелинейных преобразователей

На рисунке 2.3 представлена схема вычислений в соответствии с формулами (2.10), (2.14), определяющими алгоритм настройки параметров нейронной сети для рассматриваемого случая $K = 1$. В целях упроще-

ния схемы применяются векторно-матричные обозначения для переменных. Блок I представляет собой сеть прямого распространения входного сигнала. Блок II отражает правило эволюции матрицы $\bar{w}$ в процессе градиентного спуска. Блок III служит для вычисления матрицы частных производных $\frac{\partial}{\partial \bar{w}_{ij}} D$, $i = \overline{1, N}$, $j = \overline{0, M}$. В составе блока III звено, отмеченное знаком “*”, функционирует согласно выражению (2.13), а звено, отмеченное знаком “***”, вычисляет все парные произведения $\Delta_i p_j$ элементов входных векторов $\bar{p}$ и Δ ($i = \overline{1, N}$, $j = \overline{0, M}$).

Перед началом процесса обучения устанавливаются некоторые произвольные значения элементов матрицы $\bar{w}(0)$. Эта операция называется инициализацией нейронной сети. Обычно для этих целей используется датчик случайных чисел, распределенных равномерно на отрезке $[-c; c]$, где c — параметр, устанавливаемый пользователем. Параметр c должен быть достаточно малым. В противном случае нейрон может оказаться в зоне насыщения сигмоидальной характеристики, когда он нечувствителен к любым малым изменениям синаптических коэффициентов. Эволюция соответствующих коэффициентов прекращается, хотя их значения далеки от оптимальных. Этот эффект называется “параличом” сети.

После установки значений $\bar{w}(0)$ блоки I и III функционируют P раз в соответствии с объемом обучающей выборки, что позволяет вычислить и накопить сумму (2.14), определяющую значение $\frac{\partial}{\partial \bar{w}_{ij}} D$ во всех обучающих примерах. Этот цикл обычно называют эпохой. Далее в блоке II реализуется подстройка элементов матрицы $\bar{w}$, в результате чего формируется $\bar{w}(1)$:

$$\bar{w}_{ij}(1) = \bar{w}_{ij}(0) - \alpha \frac{\partial D(0)}{\partial \bar{w}_{ij}}.$$

Это значение устанавливается в блоке I, после чего реализуется следующая эпоха и производится следующий такт подстройки коэффициентов в блоке III. Процедура повторяется до тех пор, пока не будет достигнут заданный уровень ошибки D , которая вычисляется в соответствии с выражением (2.11) в процессе обучения сети. Если требуемый уровень по ошибке D не достигается, то это может быть обусловлено несколькими причинами. Возможно, в рамках выбранной архитектуры сети принципиально невозможно достигнуть заданной точности. Другая причина — “паралич” сети. Возможно также, что текущее значение матрицы $\bar{w}$ соответствует положению точки в области обширного “плоскогогорья” на поверхности функции $D(\bar{w})$ (эта поверхность обычно называется адаптивным рельефом), когда продвижение точки сильно замедлено. В этом случае можно увеличить значение параметра α , чтобы ускорить продвижение к “обрыву” на адаптивном рельефе. В программных пакетах, реализующих общую идею обучения, обычно применяют различные модификации градиентного спуска, ускоряющие процесс поиска минимума показателя D .

Метод обратного распространения ошибки для обучения МНС

Метод обратного распространения ошибки (error backpropagation), который является одним из самых распространенным в практических приложениях нейронных сетей, был сформулирован независимо друг от друга несколькими русскими и зарубежными учеными в 80-е годы.

Для иллюстрации излагаемого далее принципа рассмотрим следующий пример [8]. Пусть нелинейный преобразователь $F(x)$ представляет собой последовательное соединение нелинейных элементов $f_1, f_2, \dots, f_K$ (см. рис. 4). В соответствии со схемой преобразователя можно записать $F(x) = f_K(f_{K-1}(\dots f_1(x)))$.

Поставим задачу вычисления производной $\frac{d}{dx}F$. Согласно правилу дифференцирования сложной функции

$$\frac{d}{dx}F(x) = f'_K(y_{K-1})|_{y_{K-1}(x)} f'_{K-1}(y_{K-2})|_{y_{K-2}(x)} \dots f'_1(x)$$

На рисунке 2.4 показано, что этот результат формируется на выходе дополнительной цепочки, в которой последовательно соединены блоки перемножения. В этой цепочке реализуется обратное движение сигнала с использованием результата прямого распространения входного воздействия x .

Этот пример наводит на мысль о возможности вычисления частных производных, необходимых для обучения МНС по закону (2.10), с использованием известных правил дифференцирования сложной функции и реализации обратного распространения сигнала ошибки (роль F играет показатель D точности обучения сети).

В качестве первого шага рассмотрим выражение для частных производных $\frac{\partial}{\partial \bar{w}_{ij}^K} D$ функционала D по настраиваемым параметрам последнего слоя:

$$\begin{aligned} \frac{\partial}{\partial \bar{w}_{ij}^K} D &= \frac{\partial}{\partial \bar{w}_{ij}^K} \left(\frac{1}{2} \sum_{p=1}^P \sum_{m=1}^{N=N_k} (\sigma_m^p - s_m^{pK})^2 \right) = \\ &= - \sum_{p=1}^P \sum_{m=1}^{N=N_k} (\sigma_m^p - s_m^{pK}) \frac{\partial}{\partial \bar{w}_{ij}^K} s_m^{pK} \\ i &= \overline{1, N_K}, j = \overline{0, N_{K-1}}. \end{aligned} \quad (2.15)$$

Вычисление частной производной $\frac{\partial}{\partial \bar{w}_{ij}^K} s_m^{pK}$ опирается на уравнение функционирования K -го слоя МНС:

$$s_m^{pK} = f_K(s_m^{pK} = f_K \sum_{r=0}^{N_{K-1}} \bar{w}_{mr}^K s_r^{p(K-1)}, m = \overline{1, N_K}) \quad (2.16)$$

Из уравнения (2.16) следует, что

$$\frac{\partial}{\partial \overline{w}_{ij}^K} s^{pK}_m = \frac{\partial}{\partial \overline{w}_{ij}^K} s^{pK}_m = \begin{cases} 0, & m \neq i \\ f'(h_i^{pK}) s_j^{p(K-1)}, & m = i \end{cases}, \quad (2.17)$$

$$i = \overline{1, N_K}, j = \overline{0, N_{K-1}}.$$

После подстановки выражения (2.17) в (2.15) получим:

$$\frac{\partial}{\partial \overline{w}_{ij}^K} D = - \sum_{p=1}^P (\sigma_i^p - s_i^{pK}) f'_K(h_i^{pK}) s_j^{p(K-1)}$$

С использованием обозначения

$$\Delta_i^{pK} = (\sigma_i^{pK} - s_i^{pK}) f'_K(h_i^{pK}), \quad i = \overline{1, N_K}, \quad (2.18)$$

последнее выражение для частной производной преобразуется к следующему виду:

$$\frac{\partial}{\partial \overline{w}_{ij}^K} D = - \sum_{p=1}^P \Delta_i^{pK} s_j^{p(K-1)}, \quad i = \overline{1, N_K}, j = \overline{0, N_{K-1}}. \quad (2.19)$$

Перейдем к рассмотрению $(K-1)$ -го слоя с настраиваемыми коэффициентами $\overline{w}_{ij}^{K-1}, i = \overline{1, N_{K-1}}, j = \overline{0, N_{K-2}}$:

$$\begin{aligned} \frac{\partial}{\partial \overline{w}_{ij}^K} D &= - \sum_{p=1}^P \sum_{m=1}^{N=N_K} (\sigma_m^p - s_m^{pK}) f'_K(h_m^{pK}) \frac{\partial}{\partial s_i^{p(K-1)}} h_m^{pK} \frac{\partial}{\partial \overline{w}_{ij}^{K-1}} s_i^{p(K-1)} = \\ &= - \sum_{p=1}^P \sum_{m=1}^{N=N_K} (\sigma_m^p - s_m^{pK}) f'_K(h_m^{pK}) \overline{w}_{mi}^K f'_{K-1}(h_i^{p(K-1)}) s_j^{p(K-2)} = \\ &= - \sum_{p=1}^P \Delta_i^{p(K-1)} s_j^{p(K-2)} \end{aligned} \quad (2.20)$$

где использовано обозначение:

$$\begin{aligned} \Delta_i^{p(K-1)} &= \sum_{m=1}^{N=N_K} (\sigma_m^p - s_m^{pK}) f'_K(h_m^{pK}) \overline{w}_{mi}^K f'_{K-1}(h_i^{p(K-1)}) = \\ &= \left(\sum_{m=1}^{N=N_K} \Delta_m^{pK} \overline{w}_{mi}^K \right) f'_{K-1}(h_i^{p(K-1)}), \quad i = \overline{1, N_{K-1}}. \end{aligned} \quad (2.21)$$

В последнем выражении была применена формула (2.18).

Для последующих (с конца) слоев $(K-2), (K-3), \dots$ вычисления частных производных функционала D по элементам расширенной матрицы

синаптических коэффициентов в слое выполняются аналогичным образом. В итоге таких вычислений получается следующая общая формула:

$$\frac{\partial}{\partial \bar{w}_{ij}^q} D = - \sum_{p=1}^P \Delta_i^{pq} s_j^{p(q-1)}, \quad q = \overline{1, K}, \quad i = \overline{1, N_q}, \quad j = \overline{0, N_{q-1}}, \quad (2.22)$$

где

$$\Delta_i^{pq} = \left(\sum_{m=1}^{N=N_{q+1}} \Delta_m^{p(q+1)} \bar{w}_{mi}^{q+1} \right) f'_q(h_i^{pq}), \quad q = \overline{1, K-1}, \quad (2.23)$$

$$\Delta_i^{pK} = (\sigma_i^p - s_i^{pK}) f'_K(h_i^{pK}), \quad i = \overline{1, N_K}.$$

Переменные Δ_i^{pq} , $q = \overline{1, K}$, $i = \overline{1, N_q}$, получили название двойственных по отношению к потенциалам нейронов h_i^{pq} , $q = \overline{1, K}$, $i = \overline{1, N_q}$, в сети прямого распространения входного сигнала.

На рисунке 2.5 представлена схема вычислений в соответствии с формулами (2.22), (2.23). Схема содержит цепь обратного распространения, которая возбуждается сигналом ошибки $e = \sigma - s^K$. В целях упрощения обозначений в схеме опущен индекс p примера обучающей выборки. Цепь обратного распространения формирует двойственные переменные Δ^q , $q = \overline{1, K}$, являющиеся векторами размерности N_q . Знаками “*” и “**” отмечены звенья, которые осуществляют то же преобразование данных, что и в схеме на рисунке 2.3.

Для управления процессом настройки параметров МНС согласно системе уравнений (2.10) следует активизировать вычисления (см. рисунок 2.5) P раз по числу обучающих примеров и провести накопление результатов подобно тому, как это показано на рисунке 2.3 для простейшего случая однослойной сети.

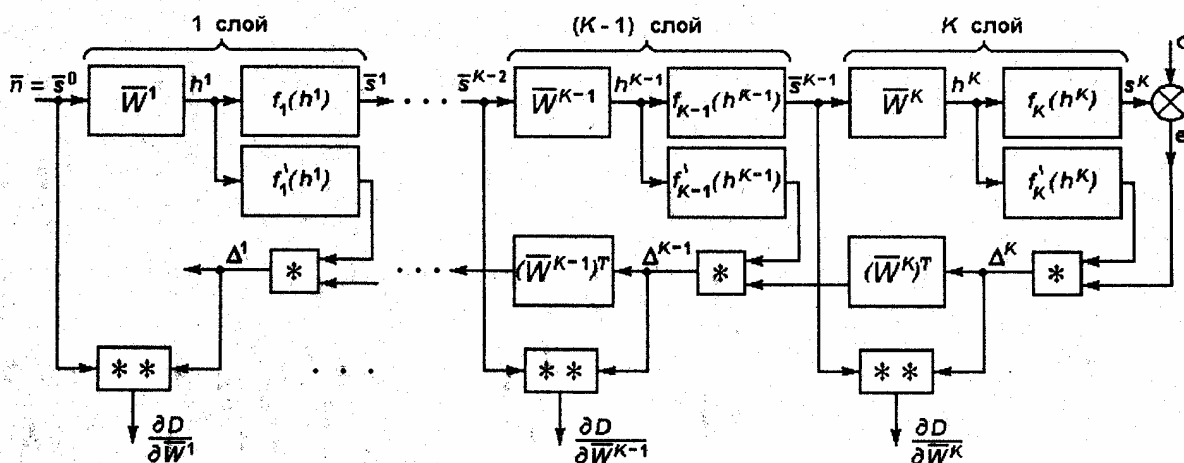


Рисунок 2.5 – Схема сетей прямого распространения и обратного распространения ошибки

Сопоставление схем прямого и обратного распространения в их полном (а не сжатом векторно-матричном) представлении показывает, что схема обратного распространения может быть построена по заданной прямой схеме путем применения к последней следующих правил:

1. Направление стрелок, указывающих прохождение сигнала, меняется на обратное.
2. Фрагменты схемы с активационными характеристиками нейронов $f_q(h_q^i)$, $q = \overline{1, K}$, $i = \overline{1, N_q}$ заменяются нелинейным преобразователем $f'_q(h_q^i)$ и блоком перемножения.
3. Матрицы синаптических коэффициентов $\overline{W}^q$, $q = \overline{1, K}$, транспонируются.
4. Сумматоры заменяются точками разветвления, а точки разветвления – сумматорами.

Сформулированные правила построения схемы формирования двойственных переменных справедливы и в том случае, когда нейронная сеть

содержит прямые связи не только рядом расположенных слоев, но и более удаленных (связь “перепрыгивает” несколько ближайших слоев).

Практические исследования показывают, что обученная МНС обладает высокой робастностью: при установке значений синаптических коэффициентов, отличающихся от оптимальных (ошибки реализации), сеть продолжает выполнять свою функциональную задачу. Даже разрыв некоторых синаптических связей (технический отказ отдельных элементов вычислительной сети) может не приводить к потере работоспособности сети (отказоустойчивость). Следует заметить, что указанные свойства проявляются только в том случае, когда нейронная сеть обладает некоторой информационной “избыточностью” по отношению к решаемой задаче.

Принципиальной является способность МНС к обобщению, то есть способность формировать “разумную” реакцию на входные воздействия, которых не было в составе обучающей выборки. Именно благодаря этому свойству МНС успешно применяется для интерполяции функций многих переменных, экстраполяции временных рядов, классификации объектов по их признакам и в других практических приложениях.

Последовательность выполнения работы

1. Определение типа решаемой задачи. В соответствии с вариантом задания (таблица 2.1) сделать заключение, что нейросеть будет применяться для решения задачи: аппроксимации функций, прогноза значений временного ряда или классификации.

Таблица 2.1 – Варианты заданий

№ варианта	Задача	№ варианта	Задача
1	Аппроксимация var1	12	Классификация var4

2	Прогноз var1	13	Аппроксимация var5
3	Классификация var1	14	Прогноз var5
4	Аппроксимация var2	15	Классификация var5
5	Прогноз var2	16	Аппроксимация var6
6	Классификация var2	17	Классификация var6
7	Аппроксимация var3	18	Классификация var7
8	Прогноз var3	19	Классификация var8
9	Классификация var3	20	Классификация var9
10	Аппроксимация var4	21	Классификация var10
11	Прогноз var4	22	Классификация var11

2. Визуализация данных в пакете MATLAB. Загрузить MATLAB (6.x) и указать путь к М-файлу лабораторной работы № 6, например, используя команду `addpath` (пример: `addpath c:\neuro\lab7`). Набрать в командной строке имя М-файла `mnnapps`. Указать тип решаемой задачи (аппроксимация функций, прогноз временных рядов или классификация). В трёх верхних полях указать пути и имена файлов с обучающей, тестовой и контрольной выборками последовательно. Визуализировать указанные выборки, используя кнопку `View`. В выпадающих списках, соответствующих каждой координатной оси, укажите название столбца, содержащего данные выборки (пример: ось X – `in1`, ось Y – `out1`, `select – in2 from 0 to 2` – построить график функции `out1(in1)` при условии, что $0 < in2 < 2$). Зарисовать графики функции в различных проекциях (для задач аппроксимации), временной ряд (для задач прогноза) или геометрическое расположение классов (для задач классификации).

3. Анализ данных обучающей выборки.

Ответить на вопросы:

в задаче аппроксимации:

- 1) имеются ли выбросные значения функции (сильно отличающиеся от соседних значений)?

- 2) наблюдается ли линейная зависимость значений функции от какого-либо аргумента?
- 3) наблюдается ли колебательная зависимость значений функции от какого-либо аргумента?

в задаче классификации:

- 1) есть ли области перекрытия классов?
- 2) наблюдаются ли вложенные классы?
- 3) сколько прямых нужно провести, для того чтобы представить границу между классами?

в задаче прогноза временных рядов:

- 1) имеются ли выбросные значения ряда (сильно отличающиеся от соседних значений)?
- 2) наблюдается ли линейный тренд?
- 3) наблюдается ли колебательная зависимость значений ряда от временных отсчетов?

Сделайте вывод о том, сколько слоев должна содержать сеть и сколько нейронов должно быть в каждом слое для оптимального решения задачи (выполнение указанной функции при точности 10% и максимального выполнения свойства генерализации).

4. Создание и обучение МНС. В пакете NNTool или из командной строки MatLab создайте новый проект. В структуру новой сети введите оцененные в п.3 параметры архитектуры. Задайте точность сети. Обучите нейросеть простым градиентным методом. Зафиксируйте значение ошибки на обучающей выборке. Сохраните результаты обработки обучающей выборки в текстовый файл.

5. Тестирование МНС. Зафиксируйте результаты обработки сетью тестовой выборки и сохраните их в текстовый файл.

6. Определение работы сети для контрольных примеров. Зафиксируйте результаты обработки сетью контрольной выборки. Сохраните результаты обработки контрольной выборки в текстовый файл.

7. Визуализация результатов обработки данных нейросетью.

Укажите в трёх нижних полях пути и имена сохранённых ранее текстовых файлов с результатами обработки сетью обучающей, тестовой и контрольной выборками последовательно. Визуализируйте указанные выборки, используя кнопку View. В выпадающих списках, соответствующих каждой координатной оси, укажите название столбца, содержащего данные выборки (см. п. 1).

8. Упрощение МНС. Сократите число синапсов МНС. Получите результаты обработки сетью обучающей, тестовой и контрольной выборок (см. пп. 5, 6). Визуализируйте полученные результаты (см. п. 7). Сделайте вывод о том, как повлияло упрощение на точность работы сети и ее способность к обобщению, используя результаты тестирования выборок и визуализацию результатов работы сети на рабочей выборке.

9. Выбор лучшей архитектуры для решения поставленной задачи. Из полученных в пп. 4 и 8 архитектур выберите наиболее удачную или предложите и обучите новую. Опишите, по каким критериям происходило рассмотрение лучшей архитектуры и что привело к созданию новой архитектуры.

Содержание отчета

1. Название лабораторной работы, цель и программа работы.
2. Графики с изображением обучающей, тестовой и контрольной выборок.
3. Выводы о структуре сети. Архитектура выбранной нейронной сети.

4. Коды MatLab формирования и обучения нейронной сети. Графики изменения ошибки обучения. Результаты тестирования нейронной сети. Результаты обработки контрольной выборки.
5. Коды MatLab формирования и обучения сокращенной нейронной сети. Графики изменения ошибки обучения. Результаты тестирования нейронной сети. Результаты обработки контрольной выборки.
6. Выводы по использованию архитектуры сети.

Контрольные вопросы к защите лабораторной работы

1. Объясните содержание задачи аппроксимации функции многих переменных на многослойной нейронной сети.
2. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче аппроксимации функции многих переменных?
3. Объясните содержание задачи прогноза временных рядов на многослойной нейронной сети.
4. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче прогноза временных рядов?
5. Объясните содержание задачи классификации данных на многослойной нейронной сети.
6. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче классификации данных?
7. Какой критерий используется при решении задачи классификации данных с помощью многослойной нейронной сети?
8. Какие правила останова процесса обучения МНС применяются в практических приложениях?
9. В чем состоит функциональная задача дополнительного блока принятия решения на выходе МНС при использовании ее в качестве

классификатора данных?

10. Почему чрезмерное увеличение числа нейронов скрытых слоев в МНС при решении задачи аппроксимации может ухудшить точностные показатели сети? Какое свойство нейронной сети имеется в виду?
11. Почему на задачи классификации данных и прогноза временного ряда распространяются те же рекомендации по выбору архитектуры МНС, что и в задаче аппроксимации функции многих переменных?
12. Какие Вы можете предложить нейросетевые решения задачи прогноза временного ряда одновременно на несколько последовательных временных тактов?
13. Предложите вариант построения нейросетевого классификатора при числе классов 4 с использованием многослойных нейросетевых классификаторов на 2 класса?

Тема 2. Инженерия знаний

Лабораторная работа № 3 «Математические модели искусственных нейронных сетей Хэмминга»

Цель работы

Изучить математическую модель нейронной сети Хэмминга

Программа работы

1. Исследовать нейронную сеть Хэмминга с сетью MAXNET прямого распространения сигнала.
2. Исследовать нейронную сеть Хэмминга с рекуррентной сетью MAXNET.
3. Составить и защитить отчет по результатам исследований, в котором должны быть приведены архитектуры исследованных нейронных сетей, результаты моделирования нейронных сетей, выводы по обобщающей способности и скорости функционирования сетей Хэмминга.

Краткие сведения из теории

В практических приложениях часто встречаются задачи обработки данных, которые относятся к классу задач распознавания образов. К подобным задачам относятся:

- распознавание рукописных букв независимо от их масштаба, угла поворота и особенностей почерка;
- диагностика заболевания пациента по данным медицинских анализов;
- определение типа сопровождаемой цели по известным характеристикам отраженного от цели радиолокационного сигнала;

- оценка уровня экологического загрязнения водоема по характеристикам отраженного лазерного луча.

Во всех перечисленных примерах исследуемый объект характеризуется совокупностью признаков, образующих вектор $x=(x_1, x_2, \dots, x_M)$. Объект может принадлежать одному из K известных классов. Так, при распознавании русских рукописных букв заданы 32 класса. На рисунке 3.1 представлена иллюстрация к задаче распознавания образов, в которой объект характеризуется двумерным вектором признаков $x=(x_1, x_2)$, заданным в прямоугольнике допустимых значений $x_1 \in [x_1', x_1'']$, $x_2 \in [x_2', x_2'']$. Пространство признаков разделено на три непересекающиеся области, каждая из которых объединяет объекты, принадлежащие одному классу. Эти области обозначены на рисунке Ω_1 , Ω_2 , Ω_3 . Сложность решения задачи распознавания образов в практических приложениях связана с тем, что границы между классами неизвестны, их уравнения не заданы и простая логическая проверка принадлежности текущего объекта x к одной из областей Ω_1 , Ω_2 и Ω_3 невозможна.

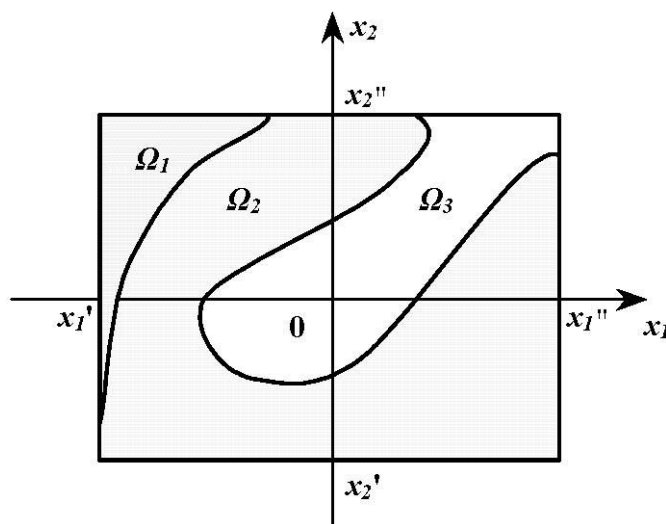


Рисунок 3.1 – Иллюстрация к задаче распознавания образов

Разные математические постановки задачи распознавания образов

связаны, как правило, с разным объемом и качеством располагаемой информации о решаемой задаче. В классической постановке задачи распознавания образов используются статистические данные относительно векторов признаков объектов в разных классах и вводится статистический критерий, позволяющий с определенной надежностью отнести объект к одному из классов. Такой подход требует предварительной статистической обработки располагаемых данных и опирается на некоторые допущения о типах распределений вероятностей, которые могут быть неточными.

В последние годы широко распространился нейросетевой подход к решению задачи распознавания образов, который опирается исключительно на экспериментальные примеры, образующие обучающую выборку. С помощью обучающей выборки осуществляется такая настройка параметров нейронной сети (синаптических коэффициентов и смещений нейронов), чтобы при предъявлении сети вектора признаков объекта x она в качестве реакции формировала указание на класс принадлежности объекта. Например, можно потребовать при настройке параметров сети, чтобы при наличии K классов она возбуждала тот из K выходных нейронов, номер которого совпадает с номером класса принадлежности объекта.

Задача распознавания образов может решаться на нейронных сетях разной архитектуры. В настоящей лабораторной работе используется простейшая модель нейронной сети, в которой каждый из нейронов выполняет четко определенную функциональную задачу и потому легко контролируется при проведении исследований.

Целью работы является изучение математической модели технического нейрона и нейронной сети, ознакомление с особенностями функционирования нейронных сетей прямого распространения и рекуррентных

нейронных сетей, исследование работоспособности нейронной сети при изменении ее внутренних характеристик.

Математическая постановка задачи

Для начального изучения модели технического нейрона и структуры нейронной сети рассматривается сеть Хемминга (Hamming). Эта сеть предназначена для распознавания класса принадлежности объекта, заданного вектором x биполярных признаков (возможные значения признаков $+1$ и -1) размерности M . Если представить $M = n_1 \cdot n_2$, то строка биполярных признаков может быть преобразована к прямоугольной матрице с n_1 строками и n_2 столбцами. Тогда возможна графическая интерпретация объекта (рисунке 3.2), в которой клетки матрицы, соответствующие признакам со значением $+1$, представлены черным цветом, а -1 – белым цветом. Предполагается, что имеются K классов, каждый из которых характеризуется своим эталонным представителем – объектом $x^{(k)}$, $k = 1, 2, \dots, K$.

Сеть Хемминга принимает на M входов биполярные признаки объекта и после выполненной обработки данных активизирует один из K выходов, который указывает на класс принадлежности предъявленного на входе объекта.

Критерием отнесения объекта x к классу является квадрат расстояния между векторами x и $x^{(k)}$, $k = 1, 2, \dots, K$:

$$R(x, x^{(k)}) = \sum_{j=1}^M (x_j - x_j^{(k)})^2, \quad (3.1)$$

где x_j – j – ый биполярный признак, $j = 1, 2, \dots, M$.

Сеть относит объект x к классу k^* , если

$$\min_{k=K} R(x, x^{(k)}) = R(x, x^{(k^*)}). \quad (3.2)$$

Раскроем выражение (3.1) для $R(x, x^{(k)})$, учитывая, что признаки являются биполярными:

$$\begin{aligned} R(x, x^{(k)}) &= \sum_{j=1}^M \left(x_j^2 + (x_j^{(k)})^2 - 2x_j x_j^{(k)} \right) = \\ &= 2 \sum_{j=1}^M (1 - x_j x_j^{(k)}) = 2 \left(M - \sum_{j=1}^M x_j x_j^{(k)} \right). \end{aligned} \quad (3.3)$$

Перейдем от критерия минимизации $R(x, x^{(k)})$ к критерию максимизации выражения $I_k = -\frac{1}{2} R(x, x^{(k)}) + 2M$, для которого достигается максимум при $k = k^*$:

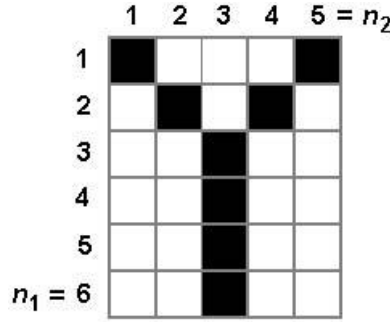


Рисунок 3.2 – Графическая интерпретация вектора признаков объекта

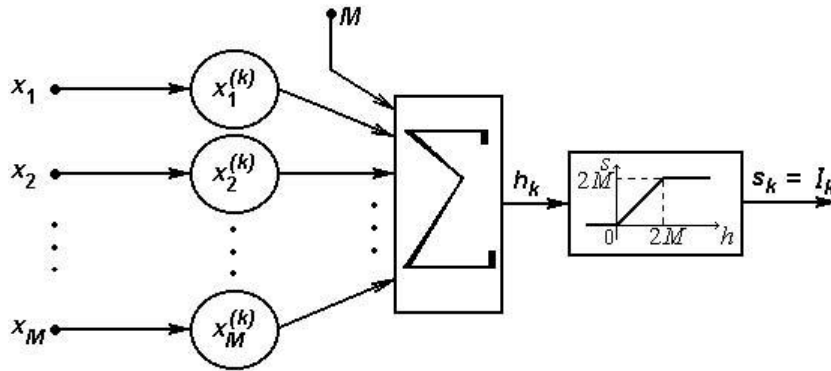


Рисунок 3.3 – Схема k -го нейрона в рабочем слое Хемминга, $k = 1, 2, \dots, K$

$$\arg \max_{k=1, \overline{K}} I_k = \arg \max_{k=1, \overline{K}} \left(\sum_{j=1}^M x_j x_j^{(k)} + M \right) = k^*. \quad (3.4)$$

Элементарный анализ показывает, что значение $R \in [0, 4M]$, поэтому $I_k \in [0, 2M]$.

Для вычисления значений I_k , $k = \overline{1, K}$, применяется рабочий слой, состоящий из K нейронов со следующими характеристиками:

$$w_{kj} = x_j^{(k)}, \quad b_k = -M, \quad k = \overline{1, K}. \quad (3.5)$$

Потенциал h_k k -го нейрона определяется с учетом (3.5) выражением:

$$h_k = \sum_{j=1}^M w_{kj} x_j - b_k = \sum_{j=1}^M x_j x_j^{(k)} + M = I_k. \quad (3.6)$$

Передаточная функция (активационная характеристика) нейронов полагается линейной с ограничениями, связанными с областью практических возможных значений потенциала $h_k = I_k$ (рисунок 3.3).

Для вычисления аргумента $k = k^*$, для которого достигается $\max_k I_k$, рабочий слой нейронов сети Хемминга дополняется группой нейронов, решающих задачу максимизации. Эта группа нейронов образует нейронную сеть, которая далее называется MAXNET. На рисунке 3.4 показана структура сети Хемминга.

В лабораторной работе рассматривается два варианта построения сети MAXNET:

- сеть MAXNET прямого распространения (feed-forward MAXNET),
- рекуррентная сеть MAXNET (recurrent MAXNET).

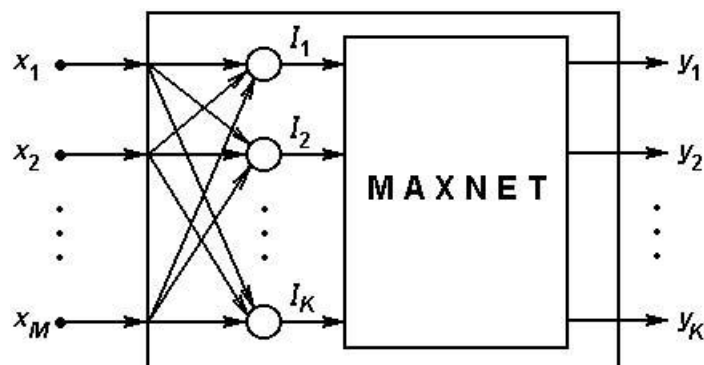


Рисунок 3.4 – Структурная схема сети Хемминга

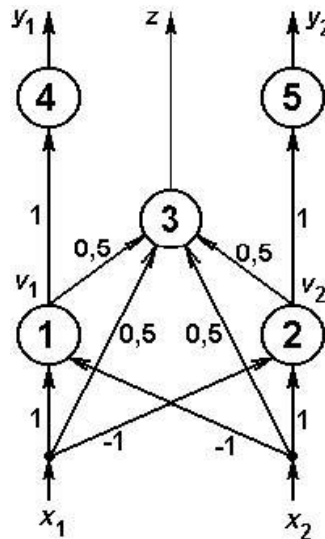


Рисунок 3.5 – Схема нейросетевого компаратора на 2 входа

Математическая модель сети feed-forward MAXNET

В основе сети MAXNET прямого распространения лежит нейросетевой компаратор на два входа (см. рисунок 3.5). Нейроны 1, 2 и 3 имеют нулевое смещение и линейную активационную характеристику с ограничениями (рисунок 3.6). Значение параметра c (уровня ограничения сверху) определяется условиями технической реализации. Предполагается, что практические значения переменных x_1 и x_2 таковы, что значения $h > c$ невозможны. Таким образом, выходы нейронов 1 и 2 могут быть представлены выражениями:

$$v_1 = \begin{cases} x_1 - x_2, & x_1 > x_2 \\ 0, & x_1 < x_2 \end{cases},$$

$$v_2 = \begin{cases} 0, & x_1 > x_2 \\ x_2 - x_1, & x_1 < x_2 \end{cases}.$$

Заметим, что значения x_1 и x_2 предполагается несовпадающими. Нейрон 3 в любой из ситуаций $x_1 > x_2$ и $x_1 < x_2$ формирует выходе $z = \max(x_1, x_2)$.

Для указания того, по какому из входов пришел максимальный сиг-

нал, служат нейроны 4 и 5. Они имеют пороговую активационную характеристику, представленную на рисунок 3.7.

Если требуется построить MAXNET на несколько входов, используется несколько компараторов, соединенных в иерархическую сеть, и выходные нейроны, играющие роль логических элементов.

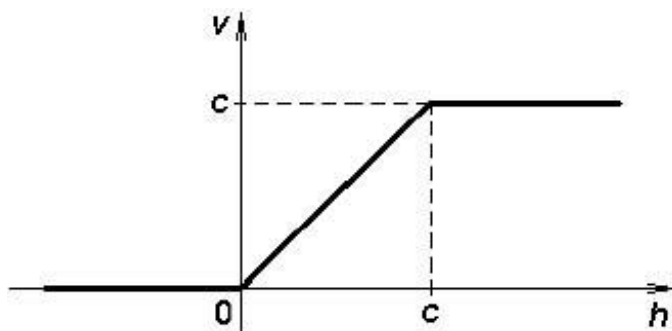


Рисунок 3.6 – Активационная характеристика нейронов 1 и 2

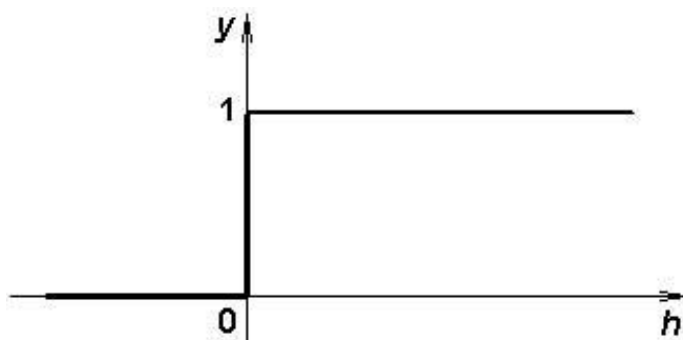


Рисунок 3.7 – Активационная характеристика нейронов 4 и 5

На рисунке 3.8 представлена архитектура MAXNET на 8 входов. Выходные нейроны, формирующие сигналы $y_1, y_2, \dots, y_8$, имеют единичные синаптические коэффициенты, нулевые смещения и пороговую активационную характеристику, представленную на рисунке 3.9. На рисунке 3.8 темными кружочками представлены нейроны, имеющие пороговые активационные характеристики и выполняющие логические функции.

Число слоев сети MAXNET прямого распространения растет пропорционально $\log_2 K$, где K – число входов. Соответственно растет и время поиска максимума, так как каждый слой обрабатывает данные за один

временной такт.

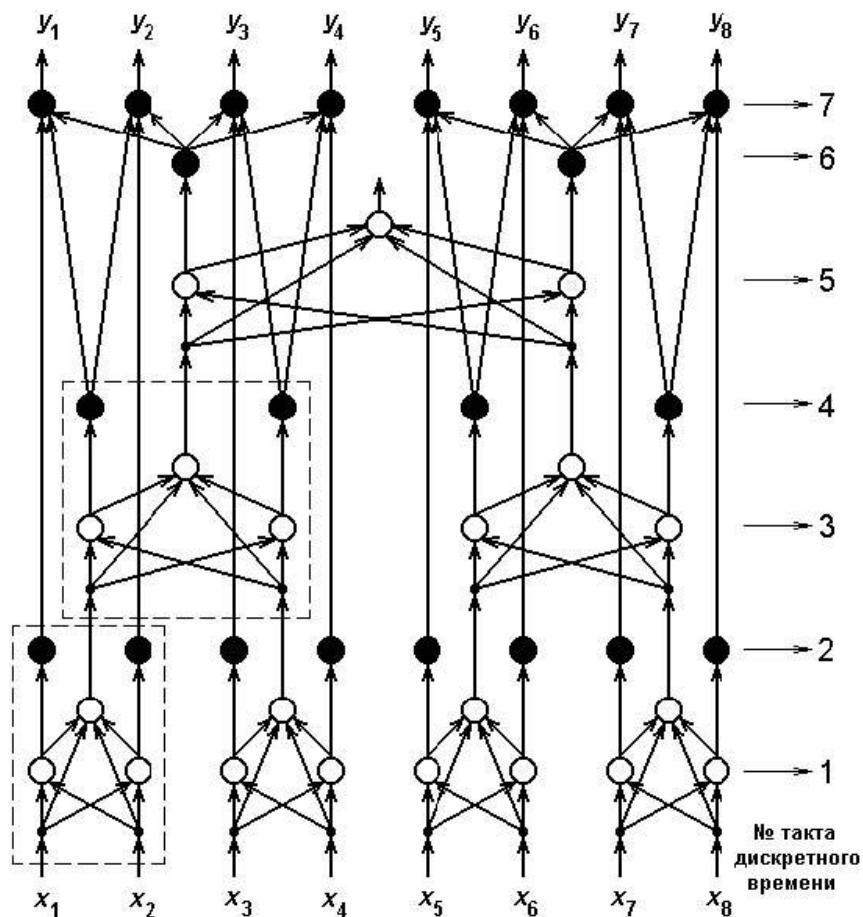


Рисунок 3.8. Схема сети MAXNET на восемь входов

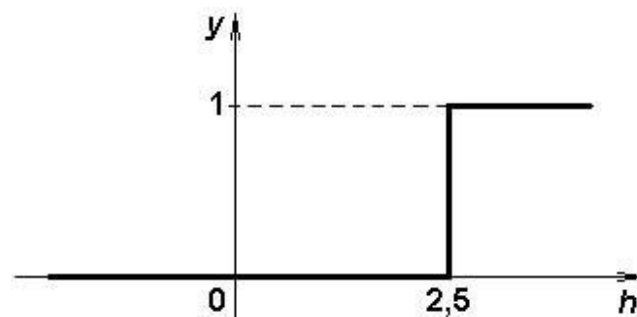


Рисунок 3.9. Пороговая активационная характеристика выходных нейронов сети MAXNET

Математическая модель рекуррентной сети MAXNET

Предполагается, что входные сигналы рекуррентной сети MAXNET положительны и не совпадают друг с другом. Обозначим их $x_1, x_2, \dots, x_K$. Рассматриваемая нейронная сеть содержит K нейронов, каждый из которых имеет линейную активационную характеристику с ограничениями (рисунок 3.6). Сеть содержит полные связи и функционирует во времени

в синхронном режиме, когда на каждом временном такте срабатывают все нейроны. Рассмотрим математическую модель i -го нейрона. Его смещение равно нулю, а синаптические коэффициенты определяются выражением:

$$w_{ij} = \begin{cases} 1, i = j \\ -\varepsilon, i \neq j, \varepsilon \leq \frac{1}{K} \end{cases}$$

Выход y_i i -го нейрона на такте $(t+1)$ дискретного времени рассчитывается по следующей итерационной формуле:

$$y_i(t+1) = f\left(y_i(t) - \varepsilon \sum_{j \neq i} y_j(t)\right), \quad (3.7)$$

где $f(h)$ – активационная характеристика нейрона.

В качестве начальных условий для эволюции рассматриваемой динамической системы рассматриваются

$$y_i(0) = x_i, i = \overline{1, M}.$$

Заметим, что

$$\left[\varepsilon \sum_{j \neq i} y_j(t) \right] \leq \varepsilon (K-1) y_{\max}(t) \leq y_{\max}(t).$$

Отсюда следует, что для нейрона, на котором достигается максимальное значение среди всех выходов $y_i(t)$, $i = \overline{1, K}$, аргумент функции f в выражении (3.7) всегда положителен и, следовательно, на следующем такте выход этого нейрона не может иметь нулевого значения. Все другие нейроны имеют более мощную отрицательную обратную связь со стороны нейрона - “победителя” с максимальным входным сигналом. Со временем выходные сигналы этих нейронов обнуляются, так как их потенциалы становятся отрицательными. Отличным от нуля остается только вы-

ходной сигнал нейрона - “победителя”. Описанная выше схема называется схемой с латеральным (боковым) торможением, в которой реализуется принцип “победитель берет все” (“winner takes all”).

В данной нейросетевой схеме расчета максимума время вычисления заранее не может быть предсказано, так как существенно зависит от входных данных. На длительность переходного процесса влияет выбор значения параметра ε : чем больше значение ε ($\varepsilon \leq 1/K$), тем быстрее он завершается.

Описание программы “Классификатор Хемминга”

Программа позволяет определять класс принадлежности объектов, представленных вектором признаков длиной 30. Число классов фиксировано и равно 3. Каждый класс задается своим эталонным представителем. Для классификации используется сеть Хемминга. В программе предоставлена возможность выбора типа модуля максимизации MAXNET: *прямого распространения* или *рекуррентный*.

Программа запускается из командной строки системы MATLAB по команде ‘hamming’. Стартовое окно программы содержит краткие сведения о решаемой задаче (см. рисунок 3.10). Управление классификатором осуществляется из главного окна программы, которое показано на рисунке 3.20. Рассмотрим важные компоненты главного окна.

Для удобства ввода и визуального восприятия векторов признаков объекта и эталонов классов предусмотрены специальные панели с матрицами черных и белых клеток размера 6×5 . Панель ввода для объекта находится в левой части экрана и обозначена «объект X». Панели эталонных представителей классов расположены в правой части экрана, пронумерованы и обозначены «эталонны классов». Для ввода векторов признаков требуется нажатиями мыши на клетки матриц (цвет клетки после од-

нократного нажатия инвертируется) установить нужные изображения объекта и эталонов, после чего включить флаг «Готово» внизу окна. В результате устанавливаются новые значения весов связей нейронов.

Тип модуля MAXNET устанавливается переключателями в нижней части экрана. Соответственно меняется изображение MAXNET в центре экрана (см. рисунки 3.11 и 3.12). Для ввода весов связей и смещения каждого нейрона рабочего слоя Хемминга предусмотрено специальное диалоговое окно (см. рисунок 3.13), которое вызывается нажатием мыши на изображение соответствующего нейрона слоя. Левая часть этого окна предназначена для ввода весов связей.

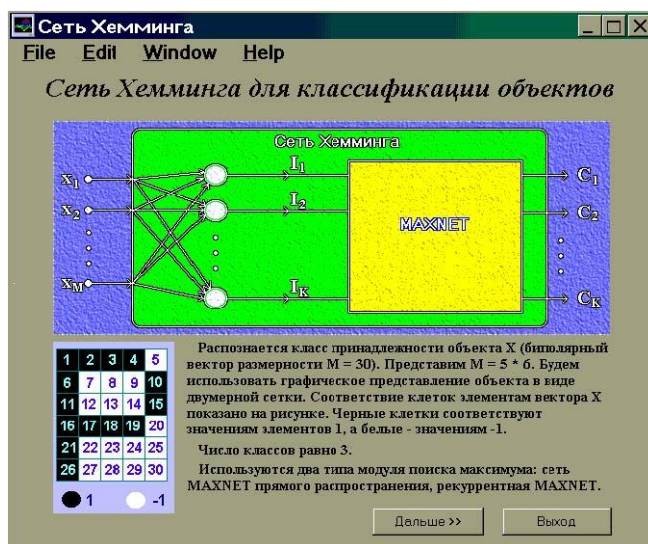


Рисунок 3.10 – Стартовое окно

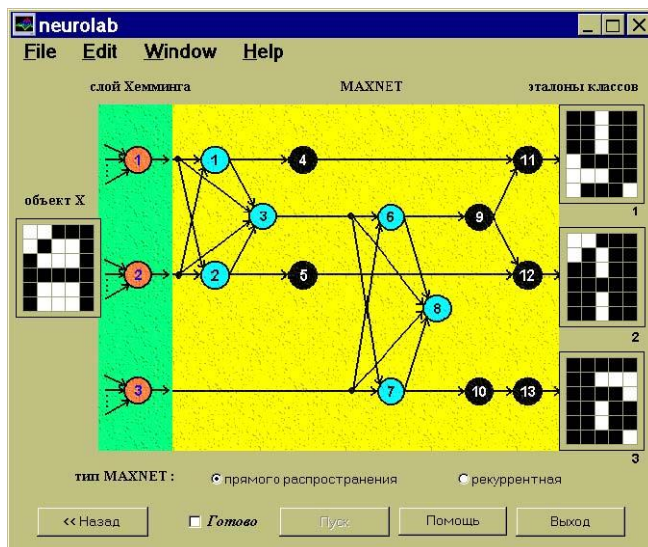


Рисунок 3.11 – Главное окно. Выбрана MAXNET прямого

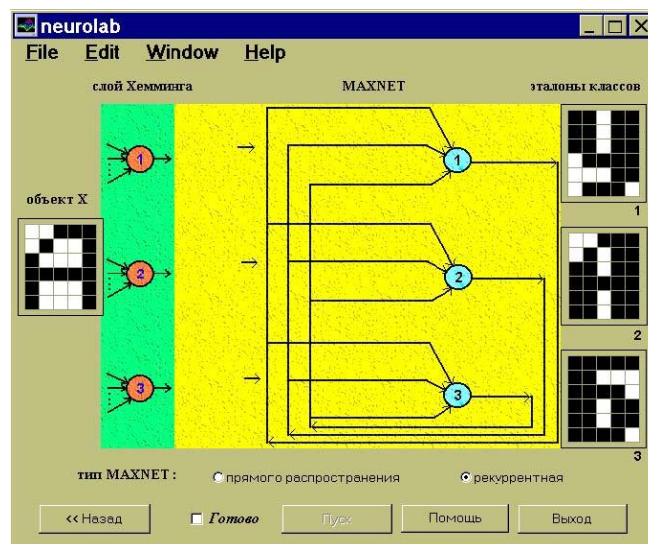


Рисунок 3.12 – Главное окно. Выбрана рекуррентная MAXNET

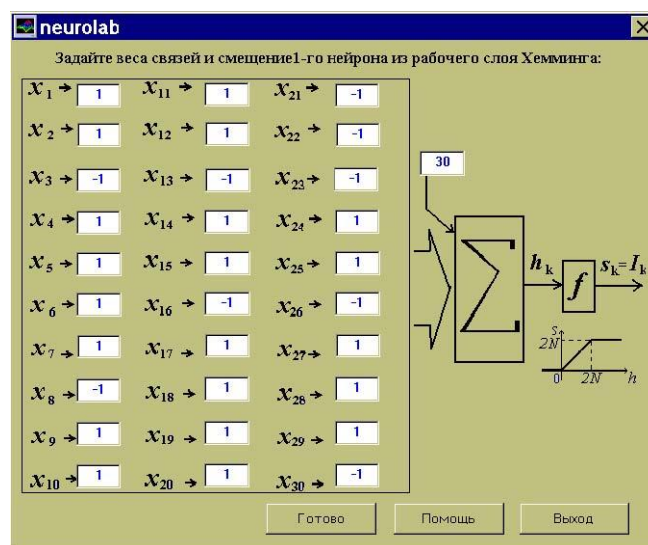


Рисунок 3.13 – Окно ввода весов и смещения 1-го нейрона рабочего слоя Хемминга

Для того чтобы просмотреть установленные или ввести новые значения весов связей сети MAXNET, нужно нажать мышью на изображение MAXNET в главном окне. В зависимости от того, какой тип MAXNET установлен в данный момент, появится окно ввода весов связей либо сети прямого распространения (см. рисунок 3.14), либо рекуррентной сети (см. рисунок 3.15).

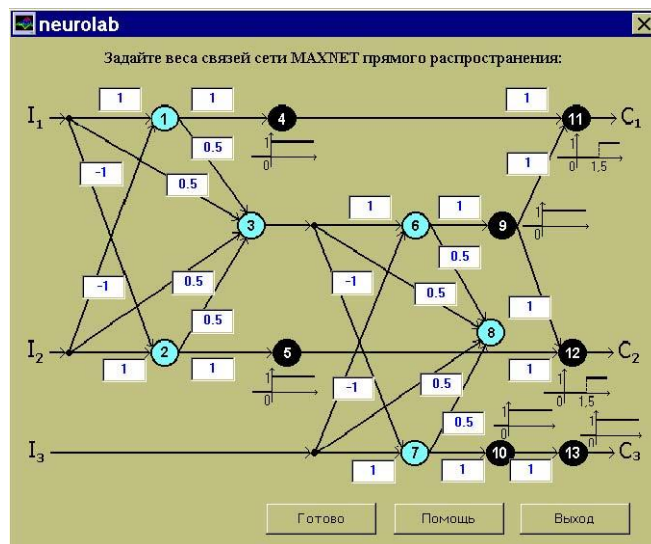


Рисунок 3.14 – Окно ввода весов связей MAXNET прямого распространения

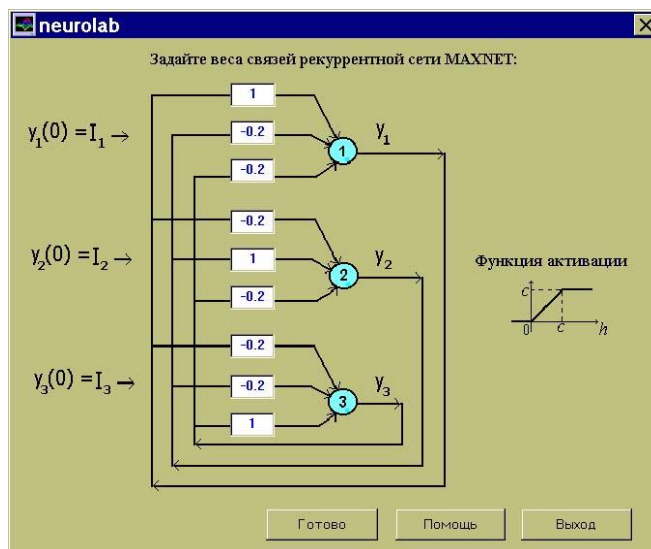


Рисунок 3.15 – Окно ввода весов связей рекуррентной MAXNET

После установки или проверки всех изменяемых параметров (с помощью описанных средств) можно запускать классификатор. Для этого нужно установить флаг «Готово». В результате этого кнопка «Пуск» внизу экрана становится доступной. Классификатор запускается по нажатию кнопки «Пуск».

Отметим, что после установки флага «Готово» автоматическая переустановка весов связей всех нейронов рабочего слоя происходит только в том случае, если за время от последней установки этого флага вносились

модификации в любой из эталонных представителей классов.

В процессе классификации на экран в области изображения сети Хемминга рядом с изображениями нейронов выводятся их выходные значения. Также на экране отображается текущий номер такта функционирования MAXNET. По окончании процесса классификации рамка панели эталонного представителя класса, к которому был отнесен предъявленный объект, выделяется цветом.

Для повторного запуска классификатора достаточно снова установить флаг «Готово» (в результате информация о предыдущей работе классификатора пропадает) и нажать кнопку «Пуск».

Последовательность действий, необходимых для классификации объекта в программе «Классификатор Хемминга», проиллюстрирована на рисунок 3.16.

1. Ввести вектора признаков эталонных представителей классов, инвертируя нажатиями мыши цвет клеток на панелях, обозначенных «эталонны классов» и помеченных соответствующими номерами.

2. Аналогичным способом ввести вектор признаков объекта, подлежащего классификации. Соответствующая панель с черно-белыми клетками помечена «объект X».

3. Для каждого из трех нейронов рабочего слоя Хемминга проконтролировать автоматически установленные веса связей и смещения (либо установить другие значения). Для этого нажатием на изображение соответствующего нейрона вызвать диалоговое окно настройки параметров этого нейрона.

4. Выбрать тип MAXNET, установив соответствующий переключатель внизу окна.

5. Проконтролировать автоматически установленные веса связей

(либо установить другие значения) сети MAXNET выбранного типа. Для этого нажатием на изображение MAXNET вызвать диалоговое окно установки весов.

6. Установить флаг «Готово». Нажать кнопку «Пуск».

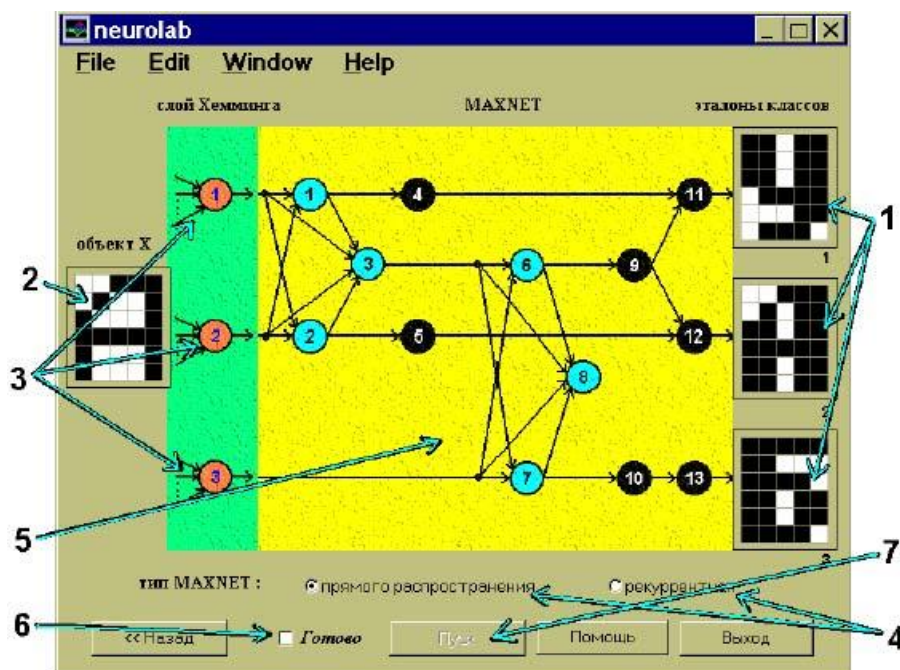


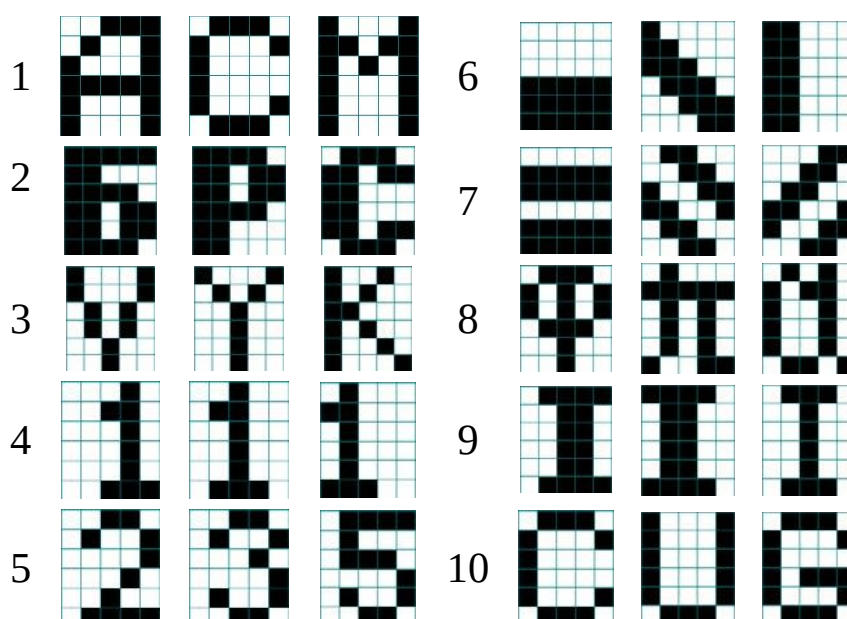
Рисунок 3.16 – Последовательность действий при классификации объекта

Последовательность выполнения работы

Классифицируемые в данной работе объекты характеризуются биполярным вектором размерности $M=30$. Для визуального представления объектов используется матрица размера 6×5 ($n_1 = 6$, $n_2 = 5$, $M = n_1 \cdot n_2$). Представленный для классификации объект может принадлежать одному из трех классов ($K = 3$). Студенту задаются изображения эталонных представителей классов на биполярной матрице размерности 6×5 .

1. Подготовить данные и осуществить запуск сети Хемминга с использованием сети MAXNET прямого распространения.
 - 1.1. Ввести эталонные изображения представителей классов.
 - 1.2. Ввести изображение текущего классифицируемого объекта.
 - 1.3. Проверить правильность установки синаптических коэффициентов всех нейронов рабочего слоя сети Хемминга.

Таблица 3.1 – Варианты заданий



1.4. Проконтролировать значения синаптических коэффициентов сети MAXNET прямого распространения.

1.5. Осуществить запуск сети. Проанализировать результат классификации и значения выходов нейронов рабочего слоя и сети MAXNET.

1.6. Повторить пункты 1.1-1.5 для разных вариантов задания текущего объекта.

2. Провести исследование возможности классификации объекта в зависимости от степени отличия его признаков от эталонного представителя класса.

Для этого проанализировать работу сети Хемминга для нескольких входных объектов, соответствующих разным уровням искажения эталонного представителя класса. Зафиксировать, при каком минимальном количестве несовпадений признаков текущего и эталонного объектов выполняется ошибочная классификация. Эксперимент провести для всех трех классов.

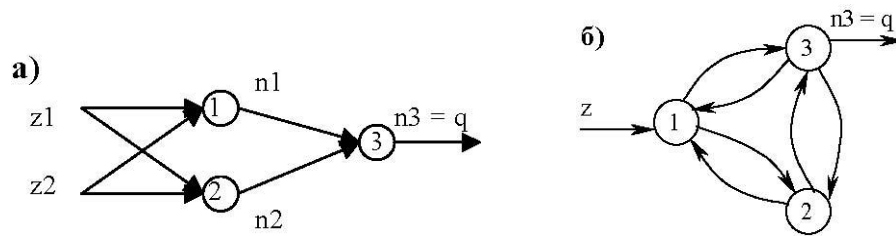
3. Провести исследование устойчивости классификации объекта при изменениях значений синаптических коэффициентов рабочего слоя и сети MAXNET. При проведении этого эксперимента могут устанавливаться произвольные действительные значения синаптических коэффициентов. Текущие объекты предполагаются близкими к эталонным представителям (отличия в двух – трех признаках). Изменением синаптических коэффициентов добиться ситуации, в которой сеть неправильно классифицирует объект, и зафиксировать уровень искажения значений синаптических коэффициентов в процентах от номинальных и число введенных искажений.
4. Подготовить данные и осуществить запуск сети Хемминга с использованием рекуррентной сети MAXNET.
 - 4.1. Ввести эталонные изображения представителей классов.
 - 4.2. Ввести изображение текущего классифицируемого объекта.
 - 4.3. Проверить правильность установки синаптических коэффициентов всех нейронов рабочего слоя сети Хемминга.
 - 4.4. Проконтролировать значения синаптических коэффициентов рекуррентной сети MAXNET.
 - 4.5. Осуществить запуск сети. Проанализировать результат.
 - 4.6. Повторить пункты 4.1-4.5 для разных вариантов задания текущего объекта.
5. Проанализировать время принятия решения о принадлежности объекта классу в зависимости от значения параметра ε и различия значений потенциалов нейронов рабочего слоя. Для этого воспользоваться соответствующей модификацией синаптических коэффициентов рекуррентной сети MAXNET и текущих анализируемых объектов.

Содержание отчета

1. Название лабораторной работы, цель и программа работы.
2. Структуры исследуемых нейронных сетей и результаты моделирования.
3. Выводы по обобщающей способности нейронных сетей и скорости функционирования.

Контрольные вопросы к защите лабораторной работы

1. Какие особенности функционирования биологического нейрона учтены при построении математической модели технического нейрона?
2. Напишите уравнение функционирования технического нейрона.
3. Что называется активационной характеристикой (передаточной функцией) нейрона? Приведите примеры.
4. Какими свойствами обладают сигмоидальные нелинейные преобразователи? Приведите примеры.
5. Что называется синаптическим коэффициентом w_{ij} математической модели нейронной сети? v_{ij} ?
6. Напишите уравнения, характеризующие динамику нейронной сети в дискретном времени (скалярная форма).
7. Напишите векторно-матричные уравнения, характеризующие динамику нейронной сети в дискретном времени.
8. Как записываются начальные условия для уравнений динамики нейронной сети?
9. Постройте математическое описание нейронной сети, представленной на рисунке а.
10. Постройте математическое описание нейронной сети, представленной на рисунке б.



11. Какая нейронная сеть называется рекуррентной?
12. Объясните особенности функционирования нейронной сети прямого распространения.
13. Какими особенностями обладает математическое описание многослойной нейронной сети?
14. Объясните принцип функционирования стохастического нейрона.
15. Какая активационная характеристика должна быть использована в детерминированном нейроне, который воспроизводит среднее значение выхода стохастического нейрона?
16. В чем состоит задача распознавания образов? Какая информация должна быть представлена об объекте при решении задачи распознавания образов?
17. Какую задачу решает сеть Хемминга?
18. В какой форме представлены характеристики объектов при решении задачи распознавания образов с помощью сети Хемминга?
19. Каким способом заданы классы при решении задачи распознавания образов с помощью сети Хемминга?
20. Сформулируйте критерий оптимальности отнесения объекта к одному из классов при применении сети Хемминга.
21. Какое содержание имеют выходные значения нейронов рабочего слоя сети Хемминга?
22. Сколько нейронов содержит рабочий слой сети Хемминга? Каковы их активационные характеристики?
23. Какими способами можно решать задачу поиска максимума среди K

значений при построении сети Хемминга?

24. Нарисуйте схему нейросетевого компаратора на два входа и объясните его работу. Какие активационные характеристики нейронов используются в схеме нейросетевого компаратора?
25. Объясните принцип построения сети MAXNET прямого распространения при произвольном числе входов с использованием нейросетевого компаратора на два входа.
26. Нарисуйте схему рекуррентной сети MAXNET. Какие начальные условия устанавливаются на сети?
27. Какой параметр рекуррентной сети MAXNET может влиять на ее быстродействие?
28. Объясните принцип работы рекуррентной сети MAXNET.

Тема 2. Инженерия знаний

Лабораторная работа № 4 «Сеть Хопфилда»

Цель работы

В лабораторной работе исследуются свойства нейронной сети Хопфилда, которая рассматривается как модель ассоциативной памяти, позволяющей восстановить объект по ограниченному набору зашумленных признаков. Сеть Хопфилда относится к классу рекуррентных нейронных сетей. В асинхронном режиме работы нейронов сеть Хопфилда из произвольного начального состояния за конечное число тактов дискретного времени приходит в состояние устойчивого равновесия (аттрактор). Чисел различных аттракторов сети определяет ее объем «памяти» (число запомненных образов).

Экспериментальное исследование свойств сети Хопфилда выполняется в нейроэмуляторе, реализованном в системе MATLAB. Изучается динамика переходного процесса, методом статистического моделирования рассчитывается число аттракторов и размеры бассейнов аттракторов. Исследуется устойчивость решений при зашумлении данных.

Программа работы

4. Исследовать процесс эволюции сети Хопфилда от начального состояния к устойчивому состоянию (динамика).
5. Определить множество устойчивых состояний сети, размер бассейнов аттракторов (аттракторы).
6. Исследовать способность сети сохранять в качестве аттракторов некоторое множество образов и релаксировать к сохраненному образу при подаче на вход этого же образа с искажениями (ассоциативная память).
7. Исследовать способность сети сохранять свойство ассоциативной памяти после искажения матрицы синаптических связей (робастность).
8. Составить и защитить отчет по результатам исследований.

Краткие сведения из теории

При организации вычислений в компьютере вызов из его “памяти” необходимых данных производится по их адресам. Имеется в виду, что требуемые данные строго локализованы в некоторой фиксированной области памяти. Человеческая память организована принципиально иначе. Вызов нужной информации обеспечивается, как правило, некоторыми ассоциациями. Здесь могут быть полезными сведения, которые уточняют особенности вспоминаемого объекта и связанные с ним события.

Таким образом, можно классифицировать устройства памяти по признакам “адресно-ориентированного поиска информации” или “поиска данных по их содержанию” (“address-oriented memory” или “content-addressable memory”). Память, в которой хранение и поиск информации основаны на ее содержании, носит название ассоциативной (associated memory). Ассоциативная память позволяет восстановить информацию в случае ее частичной потери или при значительном искажении входного запроса.

Если память позволяет восстановить поданный на ее вход искаженный неполный образ, то она называется автоассоциативной. Если при подаче на вход одного образа память реагирует извлечением другого, семантически связанного с запросом, то такая память называется гетероассоциативной (heteroassociative). Например, при отгадывании слов в кроссворде реализуются задачи автоассоциативной памяти, когда на входе заданы несколько букв слова, а память восстанавливает полное слово. При поиске марки стали по набору ее прочностных характеристик память проявляет себя как гетероассоциативная.

Сформулируем математическую задачу, связанную с функционированием автоассоциативной памяти. Предположим, что память предназначена для хранения P образов (patterns). Каждый образ представлен вектором признаков, $x^p = (x_1^p, x_2^p, \dots, x_N^p)$, $p = \overline{1, P}$. На вход автоассоциативной памяти предъявляется некоторый входной образ с набором признаков $n = (n_1, n_2, \dots, n_N)$. Требуется найти среди хранящихся в памяти такой образ x^λ , который наиболее близок к n с точки зрения евклидовой меры, т. е. для которого достигается

$$\min_{p=\overline{1, P}} H_p = \min_{p=\overline{1, P}} \sum_{i=1}^N (n_i - x_i^p)^2. \quad (4.1)$$

Если признаки являются бинарными (возможные значения 0 и 1), то показатель H_p близости образов p и x^p характеризуется расстоянием Хэмминга между двумя N –разрядными бинарными последовательностями, т. е. числом несовпадающих бинарных разрядов.

Хопфилд (Hopfield) предложил следующую идею реализации автоассоциативной памяти для образов, характеризующихся векторами бинарных признаков. Рассмотрим нейронную сеть с бинарными нейронами (пороговая активационная характеристика) и полными связями. При начальном возбуждении вектором p сеть динамически изменяет свое состояние в дискретном времени. Пусть сеть является устойчивой и за конечное число тактов приходит к одному из своих состояний равновесия. Это предельное состояние и будет представлять собой образ, ассоциированный с входным возбуждением p . Число устойчивых состояний сети (аттракторов) представляет собой число хранящихся в сети образов. Сама нейронная сеть выполняет при этом функцию ассоциативной памяти.

Хопфилд сконструировал нейронную сеть, обладающую указанными свойствами. В дальнейшем были разработаны различные модификации этой модели и статистическая интерпретация ее работы.

Рассмотрим полносвязную нейронную сеть, содержащую N нейронов по числу признаков образов. На рисунке 4.1 представлена схема такой сети. Каждый нейрон сети представляет собой биполярный элемент ($s_i = \pm 1, i = \overline{1, N}$), динамика которого в дискретном времени $t = 0, 1, 2, \dots$ описывается системой уравнений:

$$s_i(t+1) = \begin{cases} \operatorname{sgn} h_i(t), & h_i(t) \neq 0; \\ s_i(t), & h_i(t) = 0; \quad i = \overline{1, N}. \end{cases} \quad (4.2)$$

В (2.2) приняты следующие обозначения:

$$h_i(t) = \sum_{j=1}^N w_{ij} s_j(t) - b_i \text{ — потенциал } i\text{-го нейрона;} \quad (4.3)$$

b_i — смещение i -го нейрона;

$$\text{sgn}(z) = \begin{cases} 1, z > 0; \\ -1, z \leq 0; \end{cases}$$

w_{ij} — коэффициент синаптической связи j -го нейрона с i -м, $i, j = \overline{1, N}$.

В некоторых приложениях может оказаться более удобным применение бинарных нейронов с допустимыми значениями $\tilde{s}_i$, равными 0 и 1. Переход от биполярных нейронов к бинарным осуществляется с помощью линейного преобразования $s_i = 2\tilde{s}_i - 1$.

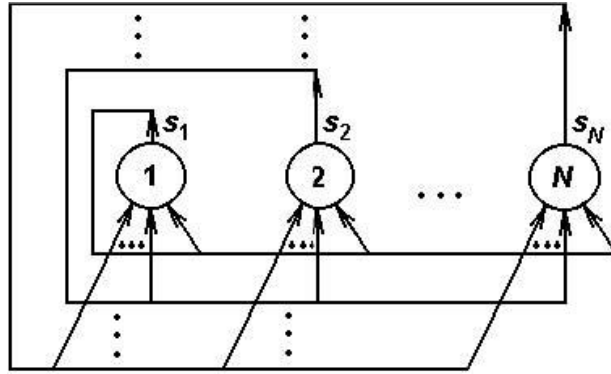


Рисунок 4.1 – Полносвязная нейронная сеть

Пусть матрица W синаптических связей размерности $[N \times N]$ является симметричной и имеет нулевые диагональные элементы (отсутствует непосредственная обратная связь с выхода нейрона на собственный вход), т.е. выполняются равенства: $w_{ii} = 0$, $i \neq j$, $w_{ij} = w_{ji}$, $i, j = \overline{1, N}$. Кроме того, положим смещения нейронов равными нулю: $b_i = 0$, $i = \overline{1, N}$.

Возможны два способа изменения состояний нейронов в соответствии с уравнениями (2.2). Первый способ предполагает, что в текущий момент времени t фиксируется вектор $s(t) = (s_1(t), s_2(t), \dots, s_N(t))$ и одновре-

менно (параллельно) изменяются состояния всех нейронов согласно уравнениям (4.2). Таким образом, за один такт дискретного времени происходит переход к новому вектору $s(t+1) = (s_1(t+1), s_2(t+1), \dots, s_N(t+1))$. Такое функционирование нейронной сети получило название синхронного.

Другой способ организации вычислений в сети состоит в последовательном изменении состояний нейронов согласно уравнениям (4.2). На каждом такте дискретности фиксируется один нейрон и вычисляется его новое состояние. Порядок опроса нейронов в такой последовательной процедуре может быть произвольным. (Хопфилд рассматривал схему случайного равновероятного выбора нейронов для расчета их новых состояний). Описанная последовательная динамика нейронной сети получила название асинхронной. В дальнейшем динамика сети полагается асинхронной, т. е. на одном такте дискретности может измениться состояние только одного нейрона.

Рассмотрим геометрическую интерпретацию состояний нейронной сети. N признаков (N нейронов) образуют N -мерное пространство. Поскольку для признаков допустимыми значениями являются -1 и $+1$, любой вектор состояния нейронной сети направлен в вершину гиперкуба, имеющего центр в начале координат и ребра длины 2, параллельные осям координат (см. рисунок 4.2). При асинхронной динамике сети вектор состояния за один такт дискретности может переместиться в одну из вершин гиперкуба, непосредственно прилегающих к текущей вершине (расстояние равно длине ребра).

Пусть заданы P образов своими векторами признаков x^p , $p = \overline{1, P}$. Определим синаптические коэффициенты, $i \neq j$, w_{ij} , $i, j = \overline{1, N}$, следующим выражением:

$$w_{ij} = k \sum_{p=1}^P x_i^p x_j^p, k > 0. \quad (4.4)$$

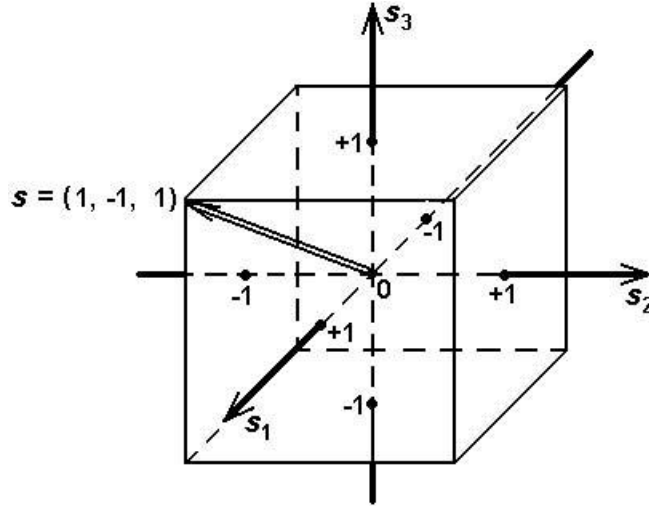


Рисунок 4.2 – Геометрическая интерпретация состояний сети Хопфилда

Динамические свойства сети не зависят от значения положительного параметра k . Это следует из того, что согласно уравнениям (4.2) состояние $s_i(t + 1)$ нейрона на следующем такте дискретного времени определяется только знаком его потенциала $h_i(t)$. Поскольку в соответствии с выражением (4.3) для потенциала $h_i(t)$ параметр $k > 0$ не влияет на его знак, этот параметр не влияет и на динамику сети в целом. В частности, если положить $k = 1/P$, то значение w_{ij} служит оценкой статистической связи i -го и j -го признаков рассматриваемых образов x^p , $p = \overline{1, P}$, для заданной выборки.

Текущее состояние динамической нейронной сети характеризуется следующим энергетическим функционалом:

$$E(t) = -\frac{1}{2} \sum_{i,j=1}^N w_{ij} s_i(t) s_j(t). \quad (4.5)$$

Исследуем, какие изменения претерпевает энергетический функционал в процессе эволюции состояния сети Хопфилда. Режим функционирования сети полагается асинхронным. Пусть на такте $(t + 1)$ проведен “опрос” k -го нейрона и его состояние в соответствии с формулами (4.2) и (4.3) изменилось:

$$s_k(t+1) = s_k(t) + \Delta s_k(t), \quad (4.6)$$

где $\Delta s_k(t) \neq 0$. Для остальных нейронов в соответствии с правилами асинхронного функционирования выполняется равенство:

$$s_i(t+1) = s_i(t), \quad i \neq k, \quad i = \overline{1, N}. \quad (4.7)$$

Рассмотрим изменение энергетического функционала:

$$\begin{aligned} \Delta E(t+1) &= E(t+1) - E(t) = \\ &= -\frac{1}{2} \sum_{i,j=1}^N w_{ij} s_i(t+1) s_j(t+1) + \frac{1}{2} \sum_{i,j=1}^N w_{ij} s_i(t) s_j(t) = \\ &= -\frac{1}{2} \sum_{i,j=1}^N w_{ij} (s_i(t) + \Delta s_i(t)) (s_j(t) + \Delta s_j(t)) + \frac{1}{2} \sum_{i,j=1}^N w_{ij} s_i(t) s_j(t). \end{aligned}$$

После раскрытия скобок в последнем выражении и приведения подобных членов получим:

$$\begin{aligned} \Delta E(t+1) &= -\frac{1}{2} \sum_{i,j=1}^N w_{ij} s_i(t) \Delta s_j(t) - \frac{1}{2} \sum_{i,j=1}^N w_{ij} s_j(t) \Delta s_i(t) - \\ &\quad - \frac{1}{2} \sum_{i,j=1}^N w_{ij} \Delta s_i(t) \Delta s_j(t). \end{aligned} \quad (4.8)$$

Согласно выражениям (4.6), (4.7) последняя сумма содержит только одно слагаемое с отличным от нуля парным произведением $\Delta s_i(t) \Delta s_j(t)$ при $i = j = k$, но при этом множитель $w_{ij} = w_{kk}$ равен нулю (диагональный элемент матрицы W синаптических коэффициентов). Таким образом, в

выражении (4.8) остаются только первые две суммы. В связи с тем, что $w_{ij} = w_{ji}$, эти две суммы совпадают и потому справедливо равенство:

$$\begin{aligned}\Delta E(t+1) &= -\sum_{i,j=1}^N w_{ij} s_j(t) \Delta s_i(t) = -\Delta s_k(t) \sum_{j=1}^N w_{kj} s_j(t) = \\ &= -\Delta s_k(t) h_k(t).\end{aligned}\tag{4.9}$$

Рассмотрим два возможных случая: $h_k(t) > 0$ и $h_k(t) < 0$ ($h_k(t) \neq 0$, т. к. в противном случае не может обеспечиваться условие $\Delta s_k(t) \neq 0$ в выражении (4.6).

Если $h_k(t) > 0$, то $s_k(t+1) = 1$. Это означает, что $s_k(t) = -1$ и $\Delta s_k(t) = s_k(t+1) - s_k(t) = 2$. Отсюда следует, что $\Delta E(t+1) = -2h_k(t) < 0$.

Если $h_k(t) < 0$, то $s_k(t+1) = -1$. Это означает, что $s_k(t) = 1$ и $\Delta s_k(t) = s_k(t+1) - s_k(t) = -2$. Отсюда следует, что $\Delta E(t+1) = 2h_k(t) < 0$.

Таким образом, изменение состояния нейрона сети в режиме ее асинхронного функционирования приводит к уменьшению энергетического функционала. В силу ограниченности снизу значения энергетического функционала $E(t)$ для конечного числа нейронов сети и приращений $\Delta E \leq 0$ через конечное число тактов энергетический функционал достигает одного из своих локальных минимумов, который является состоянием устойчивого равновесия сети.

Полученный результат и составляет содержание теоремы о конечности переходного процесса в сети Хопфилда.

В теории динамических систем $E(t)$ называется функцией Ляпунова. Состояние устойчивого равновесия нейронной сети (как и любой другой динамической системы) называется ее аттрактором. Сеть Хопфилда может иметь множество аттракторов, являющихся точками локальных ми-

нимумов энергетического функционала. Начиная свое движение из состояния $s(0)$, сеть Хопфилда “сваливается” в ближайший локальный минимум через некоторое число временных тактов. На рисунке 4.3 дана иллюстрация этого свойства. В целях упрощения различные возможные состояния сети указаны вдоль горизонтальной оси.

Обозначим $s(\infty)$ состояние устойчивого равновесия сети Хопфилда, а $h(\infty)$ — соответствующий вектор потенциалов нейронов. Согласно выражению (4.3) $h_i(\infty) = \sum_{j=1}^N w_{ij}s_j(\infty)$, $i = \overline{1, N}$. Это позволяет записать выражение для энергетического функционала в следующей форме:

$$E(\infty) = -\sum_{i,j=1}^N w_{ij}s_i(\infty)s_j(\infty) = -\sum_{i=1}^N h_i s_i(\infty). \quad (4.10)$$

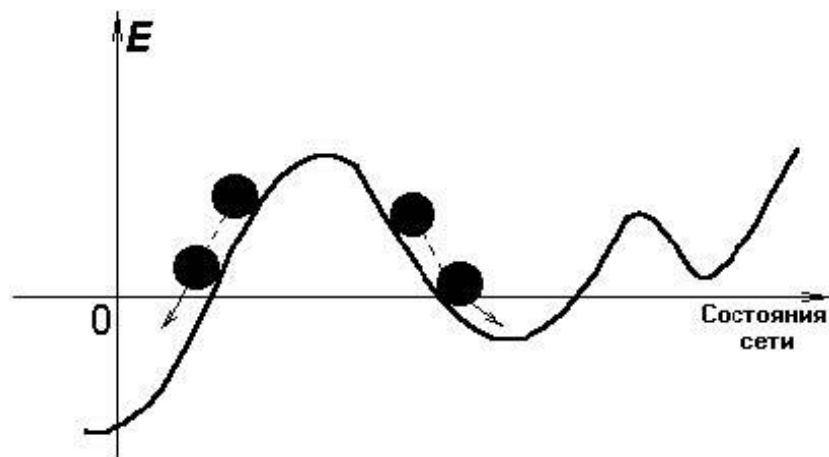


Рисунок 4.3 – Энергия сети Хопфилда

В состоянии устойчивого равновесия знак $h_i(\infty)$ согласно уравнениям динамики (4.2) совпадает со знаком $s_i(\infty)$, $i = \overline{1, N}$, или $h_i(\infty) = 0$. В противном случае наблюдались бы изменения состояний нейронов и режим не был бы установившимся. Следовательно, для всех $i = \overline{1, N}$ произведения

$h_i(\infty)s_i(\infty) \geq 0$. Применение этого неравенства к (4.10) позволяет заключить, что $E(\infty) \leq 0$.

Анализ устойчивых состояний сети Хопфилда начнем с частного случая, когда $P = 1$, т. е. в сети хранится единственный образ x^1 . В этом случае справедливо следующее выражение для синаптических коэффициентов:

$$w_{ij} = \begin{cases} 0, i = j, \\ kx_i^1 x_j^1, i \neq j, k > 0. \end{cases} \quad (4.11)$$

Проверим, является ли x^1 устойчивым состоянием нейронной сети. Предположим, что в результате эволюции сеть из некоторого начального состояния пришла в состояние x^1 в момент времени t , т. е. $s(t) = x^1$. Тогда ее состояние на следующем такте определяется выражением:

$$s_i(t+1) = \begin{cases} \text{sgn } h_i(t), h_i(t) \neq 0; \\ s_i(t), h_i(t) = 0; \quad i = \overline{1, N}, \end{cases} \quad (4.12)$$

$$\text{где } h_i(t) = \sum_{j=1}^N w_{ij} s_j(t) = k \sum_{\substack{j=1 \\ i \neq j}}^N x_i^1 x_j^1 s_j(t) = k \sum_{\substack{j=1 \\ i \neq j}}^N x_i^1 (x_j^1)^2 = \left(k \sum_{\substack{j=1 \\ i \neq j}}^N (x_j^1)^2 \right) x_i^1, i = \overline{1, N}$$

Заметим, что в связи с положительностью значения $\left(k \sum_{\substack{j=1 \\ i \neq j}}^N (x_j^1)^2 \right)$ в полученном выражении для $h_i(t)$ знак $h_i(t)$ (при $h_i(t) \neq 0$) определяется знаком x_i^1 .

В обоих случаях согласно выражению (4.12) $s_i(t+1) = s_i(t) = x_i^1, i = \overline{1, N}$. Таким образом, состояние $s(t) = x^1$ не изменяется на следующем такте времени и является, следовательно, аттрактором сети.

Анализ показывает, что аттрактором рассматриваемой сети при $P = 1$ является и инверсное состояние $(-x^1)$. Студентам предлагается самостоятельно доказать это утверждение.

Рассматриваемому в сети Хопфилда биполярному (значения $+1$ и -1) вектору признаков x^p , равно как и биполярному вектору состояний нейронов $s(t)$ можно дать простую графическую интерпретацию (см. рисунок 4.4). Рассматривается прямоугольник, содержащий $N = n_1 \times n_2$ черно-белых клеток. Каждая клетка поставлена в соответствие определенной координате вектора признаков или нейрону сети Хопфилда. Клетка зачернена, если соответствующее значение координаты вектора равно 1 , и остается белой в противном случае (значение -1). В такой графической интерпретации доказанное выше свойство сети Хопфилда может быть сформулировано следующим образом: если сеть рассчитана на хранение одного образа (изображения), то этот образ и его негатив являются аттракторами сети.

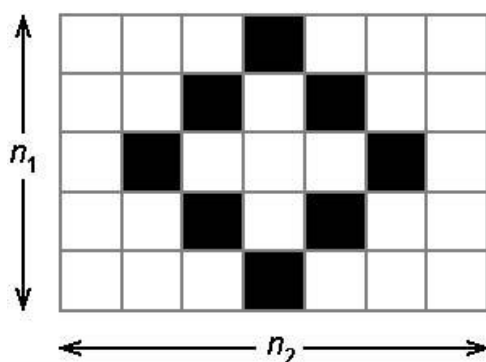


Рисунок 4.4 – Графическая интерпретация вектора состояний нейронов

Аттрактор динамической системы обладает тем свойством, что к этому состоянию сеть “сваливается”, если находится в некоторой его окрестности. Оценим ширину этой окрестности для рассматриваемого случая $P = 1$. Допустим, что в момент времени t состояние сети $s(t)$ не

полностью совпадает с аттрактором x^1 . Предположим для определенности, что различие имеет место для m координат векторов $s(t)$ и x^1 :

$$s_i(t) = \begin{cases} -x_i^1, i = \overline{1, m}; \\ x_i^1, i = \overline{m+1, N}. \end{cases} \quad (4.13)$$

Рассчитаем в этом случае состояние сети $s_i(t+1)$, $i = \overline{1, N}$. Для этого следует оценить значение потенциала $h_i(t)$, $i = \overline{1, N}$:

$$\begin{aligned} h_i(t) &= k \sum_{\substack{j=1 \\ i \neq j}}^N x_i^1 x_j^1 s_j(t) = k \left[- \sum_{\substack{j=1 \\ i \neq j}}^m x_i^1 (x_j^1)^2 + \sum_{\substack{j=m+1 \\ i \neq j}}^N x_i^1 (x_j^1)^2 \right] = \\ &= k x_i^1 [-M_{1i} + M_{2i}], \end{aligned} \quad (4.14)$$

где $M_{1i} = \sum_{\substack{j=1 \\ i \neq j}}^m x_i^1 (x_j^1)^2$, $M_{2i} = \sum_{\substack{j=m+1 \\ i \neq j}}^N x_i^1 (x_j^1)^2$. Заметим, что в зависимости от значения индекса i M_{1i} может принимать значения $(m-1)$ или m , а M_{2i} — $(N-m)$ или $(N-m-1)$ соответственно. В обоих случаях $-M_{1i} + M_{2i} \geq N-2m-1$. Пусть число различий векторов $s(t)$ и x^1 таково, что $N-2m-1 > 0$. Тогда $-M_{1i} + M_{2i} > 0$ и в соответствии с выражением (4.14) знак $h_i(t)$ определяется знаком x_i^1 . Следовательно, для любого $i = \overline{1, N}$, на такте $(t+1)$ установится значение $s_i(t+1) = \text{sgn } h_i(t) = x_i^1$.

Это означает, что за N тактов дискретного времени все нейроны гарантированно будут приведены к состояниям x_i^1 , $i = \overline{1, N}$. Таким образом, даже при 50% различии входного образа сети (состояния $s(0)$) от хранимого в сети образа x^1 (число различий m должно удовлетворять неравенству $m < (N-1)/2$) сеть дает правильную ассоциацию.

Перейдем к рассмотрению общего случая $P > 1$. Зафиксируем один из образов x^p , $p = \overline{1, P}$, на основании которых строится сеть, обозначив его x^λ . Проверим, является ли x^λ аттрактором сети. Анализ будем проводить по той же схеме, что применялась в случае $P = 1$. Допустим, что на временном такте t установилось значение $s(t) = x^\lambda$, и рассчитаем значение $s(t + 1)$. Расчет $s(t + 1)$ предполагает необходимость вычисления вектора потенциалов нейронов $h(t)$:

$$h_i(t) = \sum_{j=1}^N w_{ij} s_j(t) = k \sum_{p=1}^P \sum_{\substack{j=1 \\ i \neq j}}^N x_i^p x_j^p x_j^\lambda, \quad i = \overline{1, N}. \quad (4.15)$$

В выражении (4.15) выделим из суммы по индексу p одно слагаемое, для которого $p = \lambda$:

$$h_i(t) = k \left[\sum_{\substack{j=1 \\ i \neq j}}^N x_i^\lambda (x_i^\lambda)^2 + \sum_{\substack{p=1 \\ p \neq \lambda}}^P \sum_{\substack{j=1 \\ i \neq j}}^N x_i^p x_j^p x_i^\lambda \right] = k' (x_i^\lambda + r_i), \quad i = \overline{1, N}, \quad (4.16)$$

где $k' = k(N - 1) > 0$,

$$r_i = \frac{1}{N - 1} \sum_{\substack{p=1 \\ p \neq \lambda}}^P \sum_{\substack{j=1 \\ i \neq j}}^N x_i^p x_j^p x_i^\lambda. \quad (4.17)$$

Из выражения (4.16) следует, что знак $h_i(t)$ определяется суммой $x_i^\lambda + r_i$. Если образы x^λ и x^p , $p \neq \lambda$, $p = \overline{1, P}$, некоррелированы (ортogonalны), т. е.

$$\sum_{j=1}^N x_j^p x_j^\lambda = 0, \quad (4.18)$$

то согласно (4.17) r_i близко к нулю (отличие от нуля может быть связано лишь с исключением из суммы по индексу j в (4.17) слагаемого, для кото-

рого $i \neq j$). Таким образом, в этом случае член r_i в (4.16) в подавляющем большинстве случаев не влияет на знак $h_i(t)$ и можно считать, что знаки x_i^λ и $h_i(t)$ совпадают. Следовательно, $s(t+1)$ не будет отличаться от $s(t) = x^\lambda$, т. е. x^λ является аттрактором сети.

Рассмотрим теперь общий случай, когда свойство ортогональности (4.18) не выполняется. В этом случае уместен статистический анализ значения r_i . В связи с предполагаемым разнообразием предъявленных для обучения сети образов x^p , $p = \overline{1, P}$, можно допустить, что отдельные слагаемые $x_i^p x_j^p x_i^\lambda$ в выражении (4.17) для r_i представляют собой независимые случайные величины с равновероятными значениями $+1$ и -1 . Это означает, что математическое ожидание каждого слагаемого равно нулю, а дисперсия равна единице. Из центрированности каждого слагаемого в выражении для r_i следует, что и случайная величина r_i в целом является центрированной. Дисперсия r_i вычисляется с применением свойства независимости отдельных слагаемых, имеющих дисперсию, равную единице:

$$D(r_i) = \frac{P-1}{N-1}. \quad (4.19)$$

Для больших значений P и N можно написать следующее приближенное выражение:

$$D(r_i) \cong \frac{P}{N}. \quad (4.20)$$

Полученные статистические характеристики величины r_i позволяют построить для нее доверительный интервал $[-2.5\sigma[r_i]; +2.5\sigma[r_i]]$, где $\sigma[r_i] = \sqrt{\frac{P}{N}}$. Согласно выражению (4.16) знак $h_i(t)$ с высоким уровнем

надежности не будет отличаться от x_i^λ , если абсолютное значение r_i не превосходит единицы, т. е. границы доверительного интервала для r_i удовлетворяют неравенству:

$$2.5\sigma[r_i] = 2.5\sqrt{\frac{P}{N}} < 1.$$

Отсюда следует, что должно выполняться неравенство:

$$P < 0.16 N. \quad (4.21)$$

Выполнение условия (4.21) обеспечивает высокую вероятность того, что образ x^λ , достигнутый в сети на такте t , не изменится на следующем такте дискретности: $s(t + 1) = x^\lambda$. Иначе говоря, в этом случае x^λ является одним из аттракторов сети Хопфилда или “запомненным” сетью образом. Неравенство (4.21) можно также интерпретировать как следующее утверждение: число образов, которые может запомнить сеть Хопфилда, зависит от числа содержащихся в ней нейронов и составляет около $0.16 N$. Если образы, используемые для обучения сети, сильно коррелированы, то число устойчивых состояний (или запомненных образов) будет ниже $0.16 N$. В этом случае вполне возможно, что несколько разных образов обучающей выборки сходятся к одному и тому же аттрактору. Сами же аттракторы могут не совпадать с предъявленными сети образами.

Рассмотренные выше свойства сети Хопфилда позволяют разделить множество аттракторов любой сети на три группы. К первой группе отнесем те аттракторы, которые «притягивают» образы обучающей выборки. Именно благодаря этим аттракторам сеть Хопфилда является ассоциативной памятью. Вторую группу аттракторов образуют “негативы” (или “зеркальные отображения”) аттракторов первой группы. В третью группу включены аттракторы, которые не соответствуют ни одному примеру

обучающей выборки x^p , $p = \overline{1, P}$. Эти аттракторы называют “ложной памятью” (spurious memory).

При использовании сети Хопфилда как ассоциативной памяти аттракторы третьей группы представляют собой нежелательные состояния, которые следует подавить. В то же время некоторые авторы склонны рассматривать эти аттракторы как сгенерированные сетью новые образы, акцентируя внимание на “творческом начале”, демонстрируемом сетью [11].

Применение сети Хопфилда для кластеризации данных

Допустим, что имеется представительная выборка данных, которые должны быть разделены на группы (классы) в соответствии с принципом похожести внутри группы и различия групп. Предположим также, что априорная информация о количестве групп и их отличительных признаках отсутствует. Процедура группирования данных иначе называется кластеризацией. Имеется множество различных формализованных постановок задач кластеризации. В них предлагаются критерии качества решения и различные вычислительные процедуры для достижения цели.

Задача кластеризации может быть решена и с помощью сети Хопфилда. Поставим в соответствие группе данных один аттрактор сети Хопфилда. Этот аттрактор может рассматриваться как эталонный представитель группы, или прототип группы. Следует отметить, что прототип группы может не содержаться в обучающей выборке — “ложная память”. Число аттракторов в сети Хопфилда в такой постановке задачи кластеризации представляет собой число сформированных сетью групп.

Известен алгоритм, который приводит к решению задачи кластеризации с помощью сети Хопфилда. Этот алгоритм является многошаговым и, как установлено экспериментально, быстро сходится. На первом шаге

по данным обучающей выборки рассчитывается сеть Хопфилда и анализируются ее аттракторы. Поиск всех аттракторов осуществляется путем генерации случайных образов и использования их в качестве начального возбуждения построенной сети. При этом фиксируются все устойчивые состояния сети. Может оказаться, что некоторые аттракторы не притягивают ни одного примера из заданной обучающей выборки и поэтому не могут рассматриваться как прототипы реальных групп данных. Эти аттракторы фактически обозначают пустые кластеры.

Следующий шаг процедуры кластеризации состоит в добавлении к обучающей выборке аттракторов, соответствующих пустым классам. Основанием для такого расширения примеров для обучения является предположение, что обучающая выборка недостаточно представительна и полученные в сети аттракторы могут быть реальными данными, которые прежде не наблюдались. После расширения обучающей выборки все действия по расчету сети Хопфилда и поиску аттракторов повторяются. Как правило, число аттракторов сокращается. Уменьшается и число пустых кластеров.

Процедура расширения обучающих примеров за счет аттракторов пустых кластеров повторяется до тех пор, пока в сети не будет пустых кластеров. Число шагов процедуры зависит от числа нейронов в сети и фактической сгруппированности экспериментальных данных.

Новые обучающие примеры, сгенерированные в процессе кластеризации, обычно вливаются в группы, содержащие примеры обучающей выборки. В этом случае они могут трактоваться как возможные образы группы. В то же время допустима ситуация, когда кластер полностью состоит из добавленных примеров. Такой кластер может рассматриваться как предсказание новой возможной категории данных.

Применение сети Хопфилда для классификации объектов по заданному вектору признаков

Для построения классификатора данных, который настраивается по примерам, необходимо использовать информацию о классе принадлежности каждого из примеров обучающей выборки. Таким образом, полное множество обучающих примеров может быть представлено объединением подмножеств, каждое из которых содержит примеры одного класса. Пусть, например, решается задача распознавания буквы русского алфавита по черно-белому изображению на прямоугольной сетке. Один класс в этом примере включает в себя различные начертания определенной буквы, а общее число классов равно 32.

Свяжем с каждым классом отдельную сеть Хопфилда, синаптические коэффициенты которой рассчитываются только по тем обучающим примерам, которые принадлежат рассматриваемому классу. Разные аттракторы сети Хопфилда, настроенной на один класс, соответствуют различным модификациям объектов класса (например, различным типам начертания одной и той же буквы в задаче распознавания русских букв).

Для определения класса принадлежности некоторого объекта, который не входил в состав обучающей выборки, вектор его признаков предъявляется всем сетям классов. Каждая сеть под воздействием этого начального возбуждения эволюционирует к одному из своих аттракторов. Далее оцениваются среднеквадратические расстояния от вектора признаков объекта до аттракторов разных классов. «Побеждает» тот класс, для которого расстояние оказалось минимальным. Вместо расстояния можно использовать время переходного процесса от начального возбуждения до достижения сетью аттрактора.

Описанный способ классификации привлекателен благодаря своей наглядности и простоте модификации классификатора при добавлении новых обучающих примеров.

Последовательность выполнения работы

Данные для работы сети представляют собой набор биполярных векторов (элементы вектора могут иметь одно из двух возможных значений: $\{-1; +1\}$) одинаковой заданной длины N , которые в используемой на лабораторном занятии программе для удобства зрительного восприятия визуализируются в виде прямоугольной матрицы с заданным числом строк n_1 и столбцов n_2 ($n_1 n_2 = N$). Элементом вектора, равным -1 , ставится в соответствие белая клетка, равным $+1$ — черная (рисунок 4.5). Для создания и редактирования подобных наборов образов в программном обеспечении лабораторного занятия предусмотрен графический редактор, который сохраняет вектора и параметры n_1 и n_2 в текстовых файлах специального формата с расширением `pat`. Подробнее о программе смотрите в п. «Описание программного обеспечения».

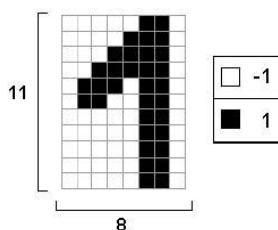


Рисунок 4.5 – Графическая интерпретация образа. Вектор из 88 элементов представлен в виде матрицы с 11 строками и 8 столбцами

Каждому из четырех указанных выше аспектов модели Хопфилда в лабораторной работе посвящено экспериментальное исследование. Рассмотрим содержание этих экспериментов.

Эксперимент «Динамика». Целью эксперимента является определение средней длительности процесса эволюции сети Хопфилда к устойчивому состоянию, длительности переходного процесса в зависимости от числа различий входа сети и устойчивого состояния, проверка свойств энергетического функционала.

На вход сети в качестве начального состояния нейронов подается некоторый образ. После этого запускается функционирование сети в соответствии с уравнениями динамики модели Хопфилда (в асинхронном режиме). При этом за один такт (цикл) *последовательно* в случайном порядке вычисляется новое состояние каждого из N нейронов сети. Порядок обработки нейронов выбирается перед каждым тактом с помощью генератора случайных чисел по равномерному закону. После каждого такта выполняется проверка идентичности достигнутого состояния сети состоянию сети после предыдущего такта. Когда эта проверка устанавливает факт достижения устойчивого состояния, динамический процесс останавливается.

Эволюция сети сопровождается изменением энергетического функционала. Текущее значение энергии вычисляется для начального состояния сети и далее пересчитывается после обработки каждого нейрона. По результатам вычислений автоматически строится график зависимости энергии сети от времени асинхронного функционирования.

Эксперимент «Аттракторы». В данном исследовании требуется найти устойчивые состояния сети Хопфилда, определить их количество, тип (истинная/ложная память), размер бассейна.

Число всех возможных состояний сети Хопфилда, содержащей N нейронов, равно 2^N . Некоторые из них являются устойчивыми, причем множество аттракторов могут составлять как те образы, по которым рас-

считана синаптическая матрица сети, так и другие состояния. Каждый аттрактор характеризуется *размером бассейна*. Бассейном аттрактора называется множество состояний сети, таких что при подаче любого из этих состояний на вход сети Хопфилда сеть в результате динамического процесса сходится к этому аттрактору. Узнать точное число аттракторов сети и размеры соответствующих бассейнов можно только перебрав множество всех возможных состояний сети (подавать каждое в качестве начального состояния и фиксировать аттрактор — достигнутое конечное состояние). Такой полный перебор потребовал бы слишком много времени на вычисления.

В лабораторном исследовании используется метод поиска аттракторов и оценивания относительного размера бассейнов, основанный на случайном поиске. Выбирается заданное число начальных состояний (случайным образом), и из каждого из них осуществляется релаксация к аттрактору. По результатам определяется число различных найденных аттракторов и относительное число попаданий в каждый из аттракторов (от общего числа начальных состояний), которое является оценкой относительного размера бассейна. Аттракторы, характеризующиеся большим значением относительного размера бассейна, считаются «более мощными» по сравнению с аттракторами, имеющими меньший бассейн («более слабые»).

Известно, что объем памяти сети Хопфилда, в том числе ложной, зависит от того, насколько коррелированы образы, по которым рассчитана сеть. Поэтому для удобства анализа результатов поиска аттракторов автоматически вычисляются коэффициенты корреляции образов (когда в сеть «записано» более одного образа).

Эксперимент «Ассоциативная память». В данном эксперименте требуется определить средний уровень искажения образа, подаваемого на вход сети, допускающий правильное восстановление исходного образа на выходе. Под искажением понимается инвертирование элементов вектора (замена -1 на 1 или наоборот), мерой искажения является расстояние Хемминга (число отличающихся элементов в исходном и искаженном векторах). Поскольку для ассоциативного восстановления разных векторов, по которым рассчитана сеть, допустимый уровень шума может быть существенно различным, требуется определить среднее допустимое искажение для образа, являющегося мощным аттрактором, и для образа — слабого аттрактора.

Эксперимент «Робастность». Требуется экспериментально проверить свойство робастности сети и определить максимальный уровень искажения синаптической матрицы (максимальную амплитуду шума), допускающий правильную работу ассоциативной памяти. Для определенности под правильной работой ассоциативной памяти в рамках данного эксперимента будем понимать способность сети правильно восстанавливать записанный образ с приблизительно 50% искажений от найденного максимально допустимого уровня в предыдущем эксперименте.

Искажение синаптической матрицы сети состоит в данной работе во внесении аддитивной помехи ξ :

$$W = W + \xi,$$

где W — матрица коэффициентов связей (размерности $N \times N$), ξ_{ij} — число, распределенное по равномерному закону на отрезке $[0, A]$, $i, j = 1, 2, \dots, N$. В таком случае уровень искажения синаптической матрицы характеризуется параметром A , который далее в лабораторной работе будем называть *амплитудой* шума. При таких условиях в результате добавления помехи

нарушается симметричность синаптической матрицы, и не выполняется условие $w_{ii} = 0, i = 1, 2, \dots, N$ (равенство нулю элементов матрицы связей по главной диагонали). Программное обеспечение позволяет искусственно поддерживать выполнение этих условий одновременно или по отдельности.

Поскольку уровень аддитивной помехи задается ее амплитудой, в программе при визуализации искажения синаптической матрицы для общности матрица характеризуется так же, как и шум, своими минимальным и максимальным элементами. Разумеется, приведенная форма задания шума и условия эксперимента являются одними из возможных для наблюдения проявлений свойства робастности.

Варианты индивидуальных заданий

Индивидуальный вариант задания и стандартный файл с образами для экспериментов выдаются непосредственно на лабораторном занятии в соответствии с номером варианта (например, номер в списке учебной группы). Стандартное множество образов предназначено для использования в 1 - 3 пп. задания (см. *порядок выполнения работы*). Для п. 4 задания требуется с помощью редактора, входящего в состав программного обеспечения лабораторного занятия, создать новый файл с множеством образов, описанным в индивидуальном задании. Ниже приведен пример индивидуального варианта:

Создать сеть Хопфилда для каждой из 4-х групп образов, выполнить для каждой сети эксперименты, указанные в описании лабораторной работы (см. *порядок выполнения работы*):

1. Цифра «3»;
2. Цифры «1», «3», «4»;

3. Цифры «1», «2», «3», «4», «5», «6», «7», «8», «9»;

4. 10 искаженных версий цифры «3».

Для сетей 1 – 3 использовать стандартное множество образов, для 4-й сети создать новое множество образов самостоятельно.

На рисунок 4.6 показано стандартное множество образов для этого варианта задания, на рис. 7 – один из возможных примеров множества образов для 4-го пункта задания.

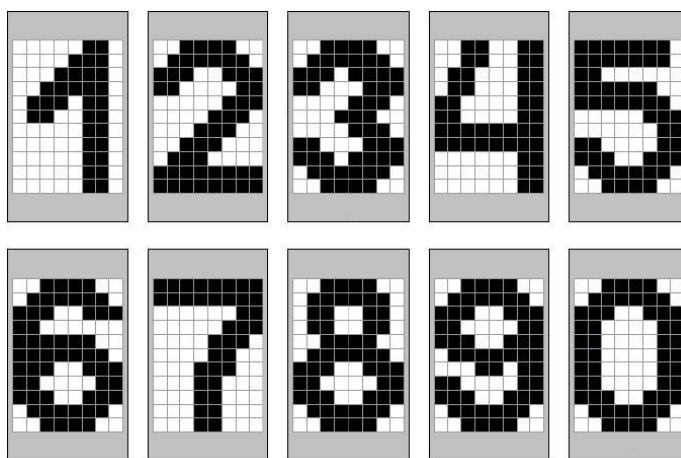


Рисунок 4.6 – Стандартное множество образов, упомянутое в приведенном примере варианта задания

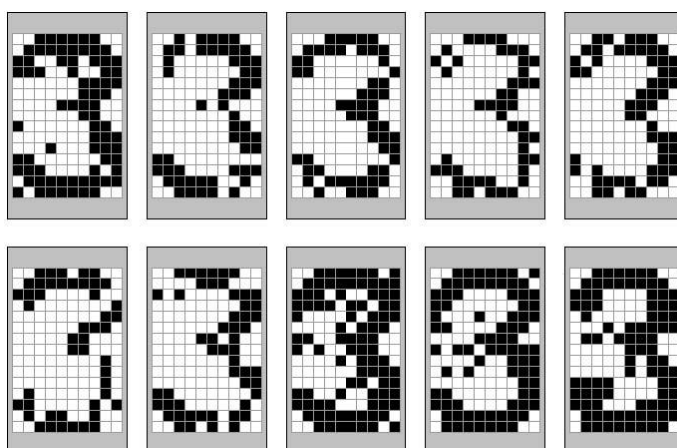


Рисунок 4.7 – 10 искаженных версий цифры «3», полученных в «Редакторе

Порядок выполнения работы

Экспериментальные исследования выполняются в следующей последовательности:

1) Создать сеть Хопфилда для одного единственного образа, указанного в 1–м п. индивидуального варианта.

1.1) Провести исследование динамики рассчитанной сети, экспериментально определить:

- является ли устойчивым состоянием записанный в сеть образ (для этого подать на вход сети в качестве начального состояния записанный образ и сравнить начальное и конечное состояние сети);

- за сколько тактов в среднем сеть от случайного начального состояния сходится к устойчивому состоянию (провести 5 – 7 экспериментов).

1.2) Провести исследование аттракторов сети, экспериментально оценить:

- количество и тип (истинная/ложная память) устойчивых состояний сети;

- относительный размер бассейнов аттракторов.

Количество начальных случайных стимулов, подаваемых на вход сети при поиске аттракторов, взять равным 1500 – 2000.

1.3) Исследовать свойство сети – ассоциативную память, экспериментально определить:

- максимальный уровень искажения записанного в сеть образа, допускающий правильное распознавание этого образа сетью (расстояние Хэмминга между исходным и искаженным образами) на основании 5 – 7 экспериментов.

1.4) Исследовать робастность сети, экспериментально определить:

- максимальную амплитуду аддитивной помехи, вносимой в синаптическую матрицу, допускающую правильную работу сети в качестве ассоциативной памяти. Уровень искажения образа поддерживается при этом постоянным и равным приблизительно 50% от максимально допустимого, полученного в п. 1.3. (ассоциативная память).

2) Создать сеть Хопфилда для группы образов, указанной в п. 2 индивидуального варианта.

2.1) Как в п. 1.1. для всех записанных в сеть образов.

2.2) Как в п. 1.2.

2.3) Как в п. 1.3. для двух записанных в сеть образов (первый из образов должен являться более мощным аттрактором, второй – более слабым).

2.4) Как в п. 1.4. для двух записанных в сеть образов (первый из образов должен являться более мощным аттрактором, второй – более слабым, те же образы, что в п. 2.3.).

3) Создать сеть Хопфилда для группы образов, указанной в п. 3 индивидуального варианта.

3.1) Как в п. 1.1. для всех записанных в сеть образов.

3.2) Как в п. 1.2.

3.3) Как в п. 1.3. для двух записанных в сеть образов (первый из образов должен являться более мощным аттрактором, второй – более слабым).

3.4) Как в п. 1.4. для двух записанных в сеть образов (первый из образов должен являться более мощным аттрактором, второй – более слабым, те же образы, что в п. 3.3.).

4) Создать сеть Хопфилда для группы образов, указанной в п. 4 индивидуального варианта (несколько искаженных версий одного образа).

Провести исследование аттракторов сети (как в п. 1.2).

Описание программного обеспечения

Лабораторная работа выполняется с помощью пакета программ «Сеть Хопфилда», разработанного для среды MATLAB 5.0. Программа «Сеть Хопфилда» позволяет загружать из текстового файла специального формата множество образов, рассчитывать синаптическую матрицу сети по одному или нескольким образам текущего множества и, после того как сеть рассчитана, можно выполнять экспериментальные исследования (описанные в п. «Содержание работы» четыре эксперимента). Кроме того, можно сохранять в текстовом файле результаты выполненных экспериментов. Для подготовки новых наборов образов или редактирования имеющихся в программу «Сеть Хопфилда» включен специальный графический редактор. Программа запускается по команде

`hopfnet`

из командной строки системы MATLAB. При запуске программы появляется окно, показанное на рисунок 4.8. В выпадающем меню в нижней части стартового окна требуется выбрать номер индивидуального варианта, после чего нажать кнопку «ОК».

В результате появляется главное окно программы, вид которого показан на рисунок 4.9. Непосредственно после запуска программы недоступен пункт «Эксперименты» главного меню и четыре кнопки в правой части окна (дублирующие подпункты меню «Эксперименты»), поскольку нейронная сеть Хопфилда пока не рассчитана.

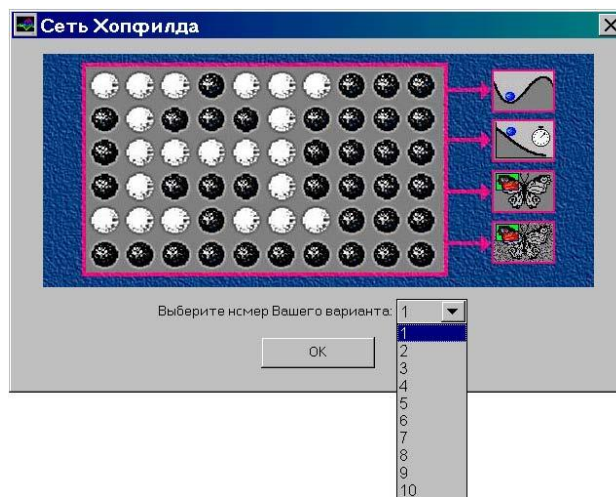


Рисунок 4.8. Это окно появляется при запуске программы
«Сеть Хопфилда»

Кратко остановимся на содержании пунктов главного меню. Меню «Файл» позволяет сохранить результаты работы и выйти из программы. Меню «Параметры» включает два подпункта: «Образы» и «Сеть», которые дублируют одноименные кнопки в левой части окна. О содержании меню «Эксперименты» уже было сказано. Меню «Помощь» помимо информации о программе включает пункт «Задание», которому соответствует одноименная кнопка в нижней части главного окна.

При запуске программы «Сеть Хопфилда» загружается текст индивидуального задания с указанным номером и соответствующий варианту стандартный набор образов. Для того, чтобы просмотреть текст задания, нужно нажать на кнопку «Задание». Появляется окно, показанное на рисунок 4.10 (текст в окне отличается для разных вариантов).

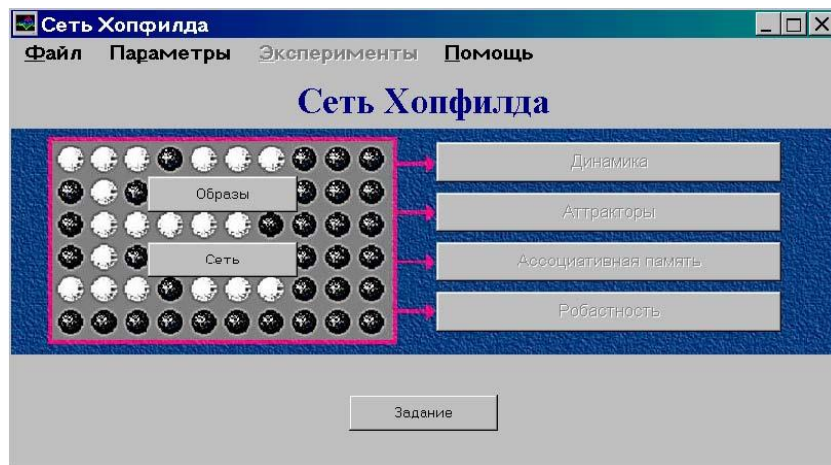


Рисунок 4.9 – Главное окно программы «Сеть Хопфилда»
непосредственно после запуска

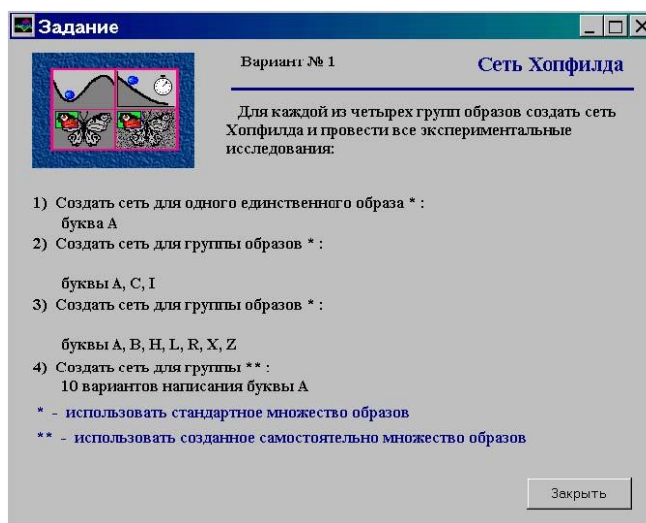


Рисунок 4.10 – Окно «Задание»

Для того, чтобы внести изменения в текущее множество образов или загрузить новое, нужно нажать на кнопку «Образы» главного окна или выбрать пункт «Образы» меню «Параметры». Появляется окно, показанное на рисунке 4.11. Центральная часть окна занята изображениями образов из текущего набора (в виде черно-белых диаграмм). Под каждым изображением выведено имя соответствующего образа. Однократное нажатие мыши на диаграмме устанавливает соответствующий образ текущим, пространство вокруг образа выделяется цветом. В нижней части окна «Образы» выводится информация о размерности образов и количе-

стве образов в текущем наборе. Кнопка «Открыть файл» предназначена для загрузки нового множества образов из файла с расширением `pat`. Для того, чтобы вернуться к главному окну, загрузив новое множество образов, требуется нажать кнопку «Применить».

При нажатии на кнопку «Редактор» в окне «Образы», запускается программа «Редактор образов», вид главного окна которой показан на рисунке 4.12.

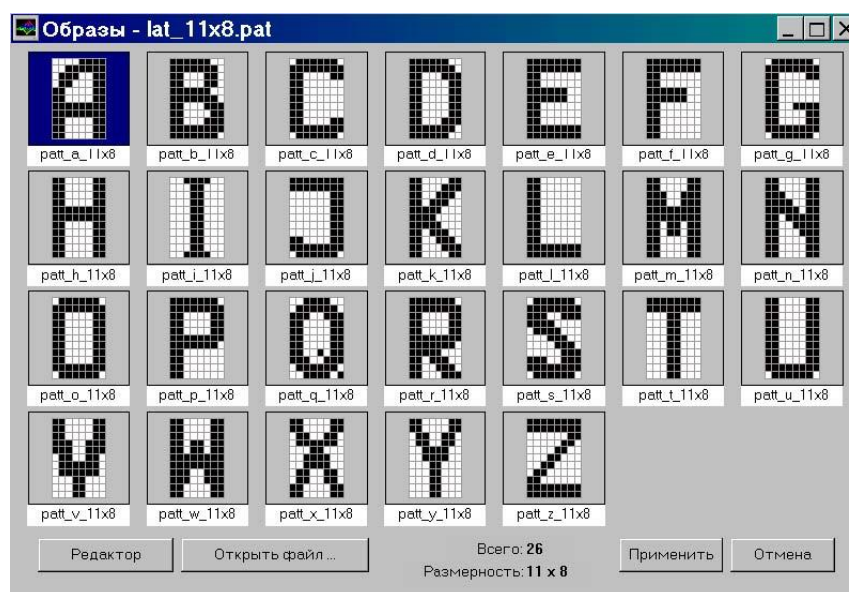


Рисунок 4.11 – Окно «Образы»

Программа «Редактор образов» является частью программного пакета «Сеть Хопфилда» и работает в среде MATLAB 5.0. Программа предназначена для создания и редактирования наборов биполярных образов, которые сохраняются в текстовых файлах специального формата. Все образы набора (множества) должны иметь одинаковую размерность, т. е. одинаковую длину вектора признаков и одинаковое число строк и столбцов для графического представления в виде черно–белой диаграммы.

Как видно на рисунке 4.12, один из образов набора изображен в центре экрана. Этот образ называется «текущим». В результате однократного

щелчка кнопки мыши на клетке диаграммы текущего образа цвет этой клетки устанавливается выбранным из «палитры» (белый или черный) в левом нижнем углу главного окна «Редактора образов». На рисунке 4.12 видно, что текущим цветом является черный. Для выбора цвета рисования нужно щелкнуть кнопкой мыши на черном или белом прямоугольнике палитры. Операции редактирования применяются к текущему образу. Для того, чтобы сделать текущим другой образ из набора используйте список в правой части окна. Заголовок окна начинается с имени файла из которого загружено множество образов. В нижней части экрана выводится информация о размерности и количестве образов в наборе.

Рассмотрим главное меню программы «Редактор образов». Меню «Файл» включает команды «Новый», «Открыть ...», «Сохранить», «Сохранить как ...» и «Выход». Команда «Новый» предназначена для создания нового множества образов. При этом требуется в специальном диалоговом окне задать размерность (число строк и столбцов) нового множества образов. Команда «Открыть ...» используется для того, чтобы открыть имеющийся файл с множеством образов. Команды «Сохранить» и «Сохранить как ...» предназначены для сохранения текущего множества образов под текущим и новым именем соответственно.



Рисунок 4.12 – Главное окно программы «Редактор образов»

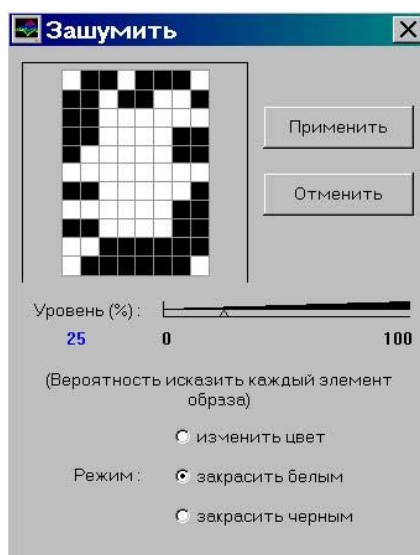


Рисунок 4.13 – Окно «Зашумить»

Меню «Образ» (рисунок 4.12) главного окна программы «Редактор образов» содержит подпункты «Обновить», «Добавить ...», «Удалить» и «Переименовать». Команда «Обновить» должна быть выполнена для сохранения результата редактирования перед сменой текущего образа. Команда «Добавить ...» предназначена для добавления нового образа в набор. При этом в появляющемся специальном окне нужно задать имя добавляемого образа. Команда «Удалить» используется для удаления текущего образа из набора. С помощью команды «Переименовать» можно изменить имя текущего образа.

Меню «Правка» включает команды редактирования «Инвертировать», «Закрасить черным», «Закрасить белым», «Зашумить» (дублируются кнопками главного окна «Редактора образов», расположенными слева) и команды «Отменить», «Повторить» (результат последней выполненной по отношению к текущему образу команды редактирования). Команды редактирования меняют цвет клеток диаграммы с белого на черный или наоборот (и, соответственно, значения элементов вектора признаков с -1

на +1 или с +1 на -1). Команда инвертировать меняет цвет всех клеток диаграммы. Команды «Закрасить черным» и «Закрасить белым» устанавливают один и тот же цвет для всех клеток. По команде «Зашумить» появляется специальное окно, показанное на рисунке 4.13.

Окно «Зашумить» предназначено для автоматического внесения искажений в текущий образ. В левой верхней части окна изображается искаженный результат, на который будет заменен текущий образ «Редактора образов» после нажатия кнопки «Применить». В центре имеется шкала с ползунком, который управляется мышью, для указания уровня зашумления в процентах. Установленный процент искажения выводится слева от шкалы (на рисунке 4.13 уровень шума равен 25%). С помощью переключателя выбирается один из трех режимов искажения: изменить цвет, закрасить белым, закрасить черным (на рисунке 4.13 выбран режим «закрасить белым»). Автоматическое искажение выполняется по такому закону: каждая клетка диаграммы изменяется в соответствии с выбранным режимом с вероятностью, равной заданному уровню шума.

При выходе из «Редактора образов» появляются диалоговые окна с предложением сохранить в файле отредактированное множество образов и загрузить это множество в программу «Сеть Хопфилда».

Для создания (расчета) сети Хопфилда, над которой можно проводить экспериментальные исследования, предназначено окно «Сеть» (см. рисунок 4.14), которое вызывается по нажатии одноименной кнопки главного окна программы «Сеть Хопфилда» или по команде «Сеть» меню «Параметры». Основную часть окна занимают черно-белые изображения образов, слева имеется список имен всех образов текущего множества.

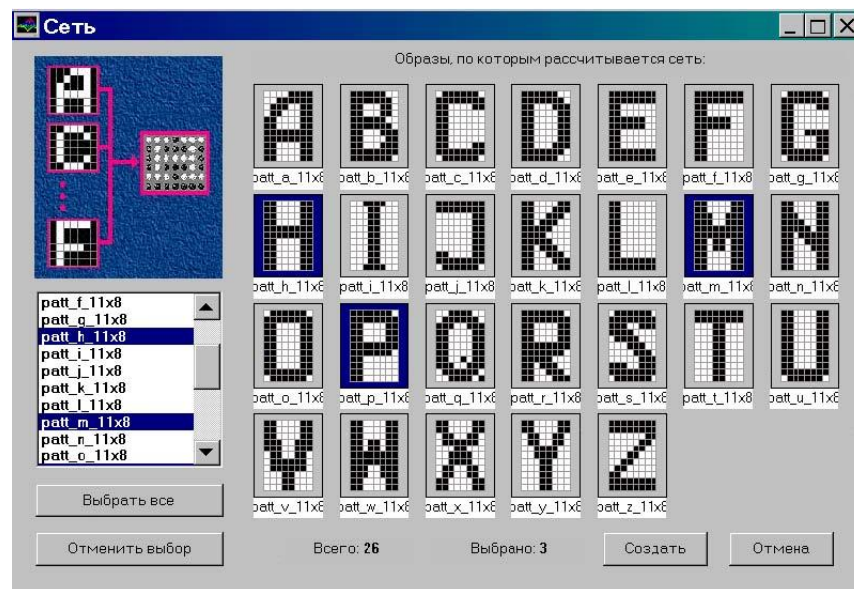


Рисунок 4.14 – Окно «Сеть»

Образ выделяется цветом фона при нажатии мыши на изображении или при указании в списке. Общее число образов и число выделенных выводится внизу окна. Для расчета сети нужно выделить нужные образы и нажать кнопку «Создать». В окне, изображенном на рисунке 4.14, выбраны три образа, по которым будет рассчитана синаптическая матрица сети. Последующие иллюстрации к описанию программы «Сеть Хопфилда» соответствуют именно этой сети.

На рисунке 4.15 показано окно эксперимента «Динамика» сразу после открытия. Для выполнения эксперимента нужно задать начальное состояние сети и выбрать режим просмотра, после чего нажать кнопку «Пуск». Начальное состояние может быть либо случайным вектором, сгенерированным автоматически, либо одним из тех векторов, по которым рассчитана сеть (способ задания начального состояния нужно указать с помощью переключателя). Предусмотрено три режима просмотра: непрерывный, пауза после каждого такта, пауза после обработки каждого нейрона. Вектор начального состояния сети можно редактировать щелчком мыши на клетках диаграммы (в режиме инвертирования). Как видно

на рисунке, заданному начальному состоянию соответствует значение энергии сети 0,15909. На рис. 16 показано окно «Динамика» после завершения переходного процесса в сети Хопфилда. В центре изображено конечное состояние сети, число тактов (равно трем), за которое оно было достигнуто, соответствующие значение энергетического функционала ($-80,5$) и график зависимости энергии сети от времени. На графике по вертикальной оси отложено значение энергии, а по горизонтальной — номер обработанного нейрона. Для того, чтобы снова наблюдать переходный процесс, нужно нажать кнопку «Повторить эксперимент».

На рисунке 4.17 представлено окно эксперимента «Аттракторы». Поскольку сеть рассчитана более, чем по одному образцу (по трем образам, соответствующим буквам латинского алфавита «Н», «М» и «Р»), имеет смысл при анализе результатов эксперимента учитывать корреляцию образов.

В левой части окна для пары образов, выбранной из списка отображается коэффициент корреляции. Единственным начальным условием эксперимента «Аттракторы» является число начальных состояний, из которых осуществляется поиск. Поиск начинается по нажатии кнопки «Пуск». В программе предусмотрено средство для принудительной остановки процесса поиска.

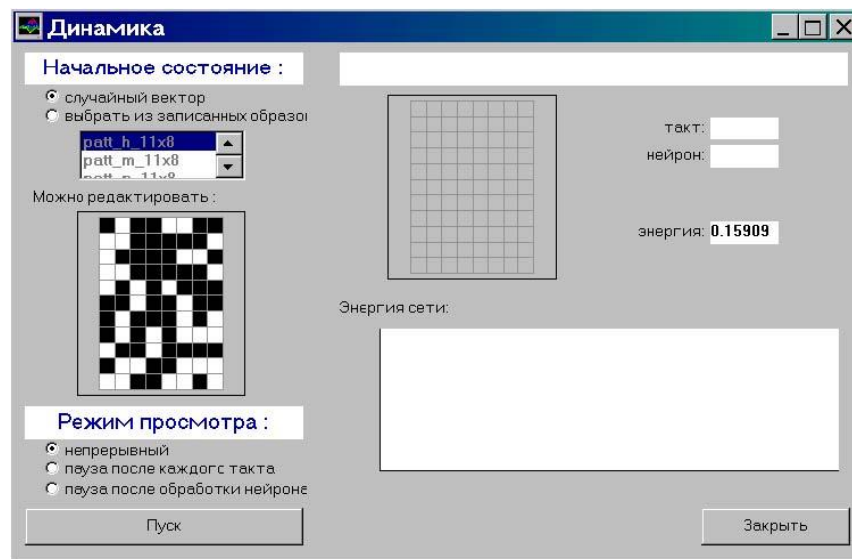


Рисунок 4.15 – Окно «Динамика»

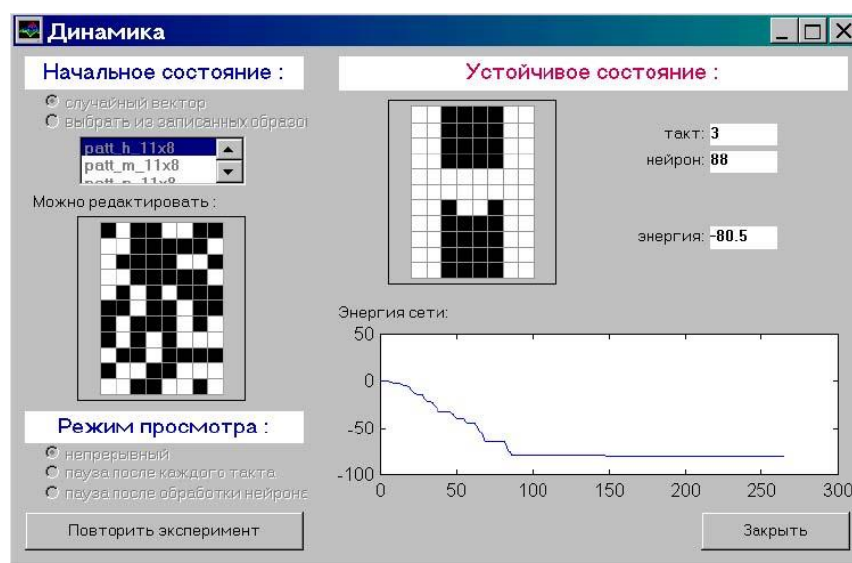


Рисунок 4.16 – Окно «Динамика» после завершения переходного процесса в сети Хопфилда

На рисунке 4.18 показан результат поиска аттракторов. Вверху окна отображено число найденных устойчивых состояний и фактическое число начальных состояний, которые проверялись во время поиска. В центре окна находится круговая диаграмма оценки относительных размеров бассейнов. Синим цветом выделен сектор, соответствующий аттрактору, изображенному справа от круговой диаграммы и выделенному в списке справа. Под изображением аттрактора выведена оценка относительного

размера бассейна с точностью до двух знаков после запятой. Для того, чтобы просмотреть другой найденный аттрактор, нужно либо один раз щелкнуть мышью на изображении одного из секторов круговой диаграммы, закрашенных серым цветом, либо выбрать другой пункт в списке найденных аттракторов. Найденные аттракторы можно сохранить в текстовом файле на диске в формате набора образов с расширением `pat`. Для этого предназначена кнопка «Сохранить аттракторы ...». В появляющемся диалоговом окне можно выбрать один из трех вариантов сохранения: добавить к текущему множеству образов, добавить к другому множеству образов, сохранить в отдельном файле.

Окно эксперимента «Ассоциативная память» показано на рисунке 4.19. Начальным условием эксперимента является начальное состояние сети, которое задается следующим образом. Сначала нужно один из образов выбрать в списке записанных в сеть (в левой части окна). Этот образ изображается на диаграмме в центре окна. Это изображение можно редактировать мышью в режиме инвертирования и с использованием кнопок, расположенных над диаграммой.

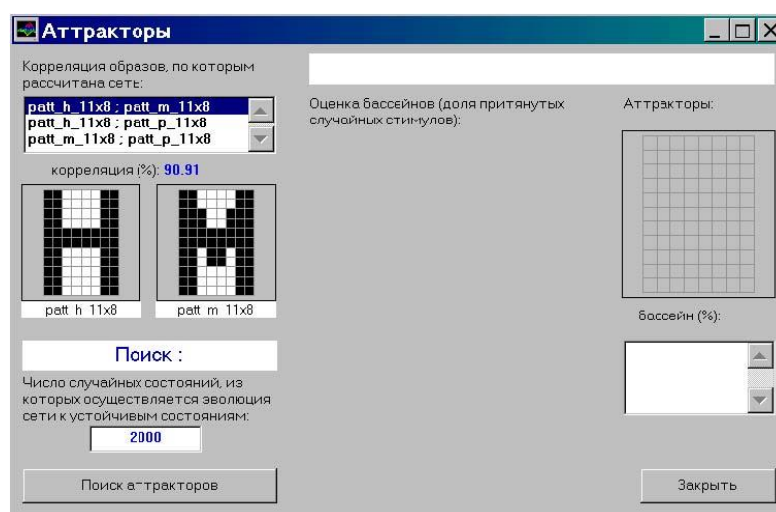


Рисунок 4.17 – Окно «Аттракторы»



Рисунок 4.18 – Окно «Аттракторы». Результаты поиска

Эти кнопки функционально эквивалентны кнопкам «Редактора образов». По мере изменения исходного образа пересчитывается и выводится в нижней части окна расстояние Хемминга между исходным вектором и текущим состоянием (на рисунке 4.19 оно равно 16). Когда начальное состояние задано, следует нажать кнопку «Пуск». На рисунке 4.20 показан вид окна после срабатывания ассоциативной памяти. В правой части окна изображено конечное состояние сети. Видно, что оно совпадает с исходным образом. Это считается правильным распознаванием и диаграмма конечного состояния выделяется зеленым цветом фона. На рисунке 4.21 показан результат другого эксперимента с тем же исходным образом. Видно, что начальное состояние сильнее отличается от исходного образа (расстояние Хемминга равно 28). Конечное состояние не совпадает с исходным вектором. Неправильность распознавания образа подчеркивается красным цветом фона диаграммы конечного состояния сети.

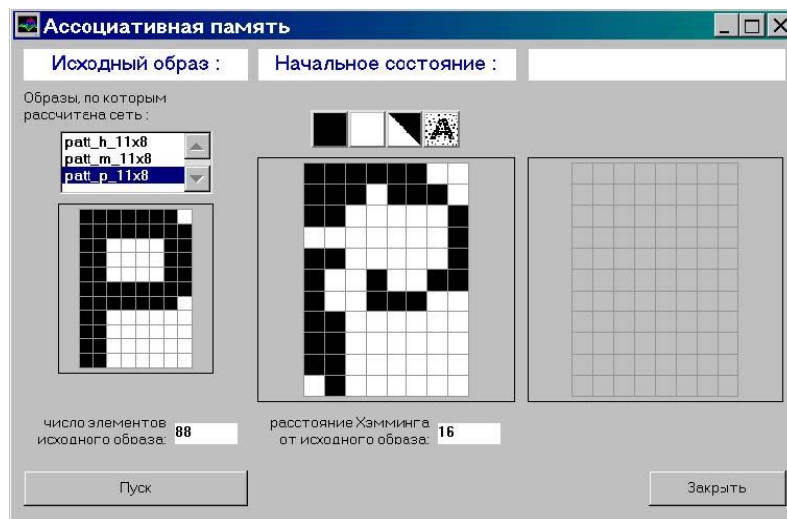


Рисунок 4.19 – Окно «Ассоциативная память»

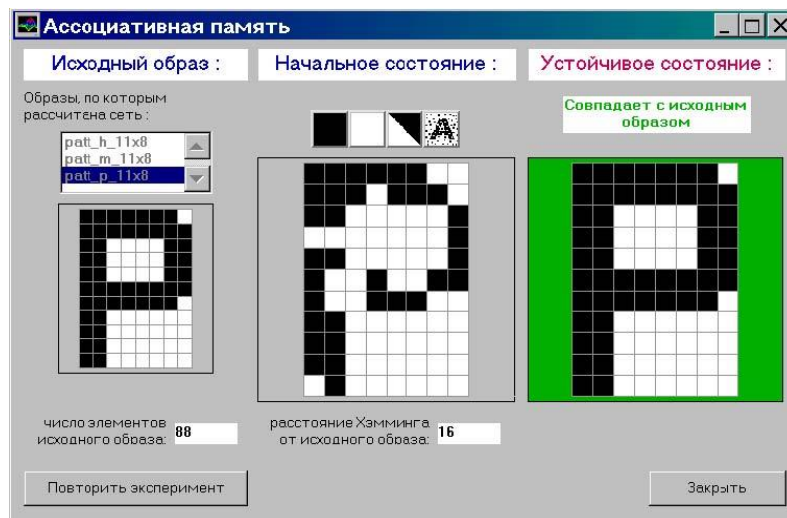


Рисунок 4.20 – Окно «Ассоциативная память».

Правильное распознавание

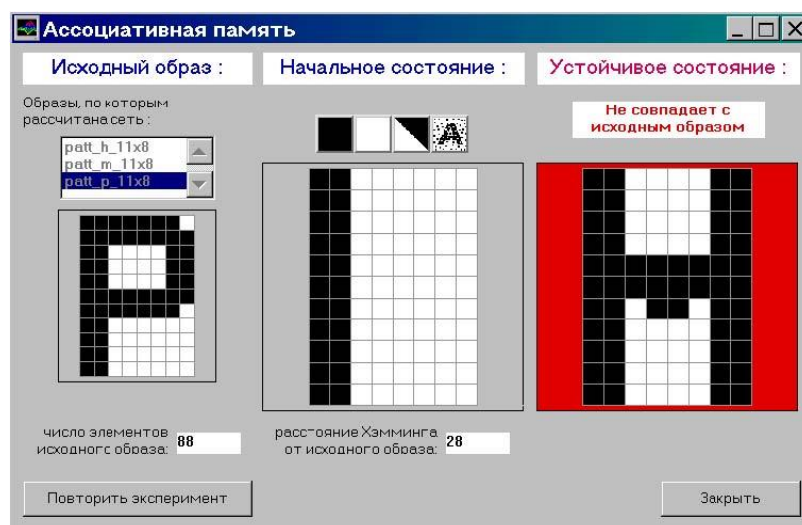


Рисунок 4.21 – Окно «Ассоциативная память».

Неправильное распознавание

На рисунке 4.22 показано окно эксперимента «Робастность». Правая часть окна почти совпадает функционально с элементами окна «Ассоциативная память». Точно так же, как и в предыдущем эксперименте, нужно выбрать исходный образ из списка и задать начальное состояние сети, зашумив исходный образ. Левая и центральная часть окна предназначена для задания параметров автоматического искажения синаптической матрицы нейронной сети. Итак, начальными условиями эксперимента являются входной образ и текущая матрица коэффициентов связей. Для искажения синаптической матрицы нужно выполнить следующие действия: задать амплитуду шума (в редактируемой строке), выставить находящийся рядом флаг «Применить», выставить желаемым образом флаги «сохранять симметричность» и «сохранять нули по главной диагонали», нажать кнопку «Зашумить исходную». Результаты этих действий изображаются на условной диаграмме в виде прямоугольников. Вид окна «Робастность» с заданными условиями эксперимента приведен на рисунке 4.23. Теперь можно нажать кнопку «Пуск». Результат срабатывания сети выводится в правом нижнем углу окна и интерпретируется аналогично эксперименту «Ассоциативная память». На рисунке 4.24 и 4.25 показаны результаты распознавания для различных начальных состояний.

Начальные условия и результаты всех экспериментов для текущей сети сохраняются в процессе работы с программой. Эти данные могут быть записаны в текстовый файл на диске по команде «Сохранить последний результат» или «Сохранить все результаты» меню «Файл» главного окна. В первом случае в файл записывается последний результат каждого из четырех экспериментов.

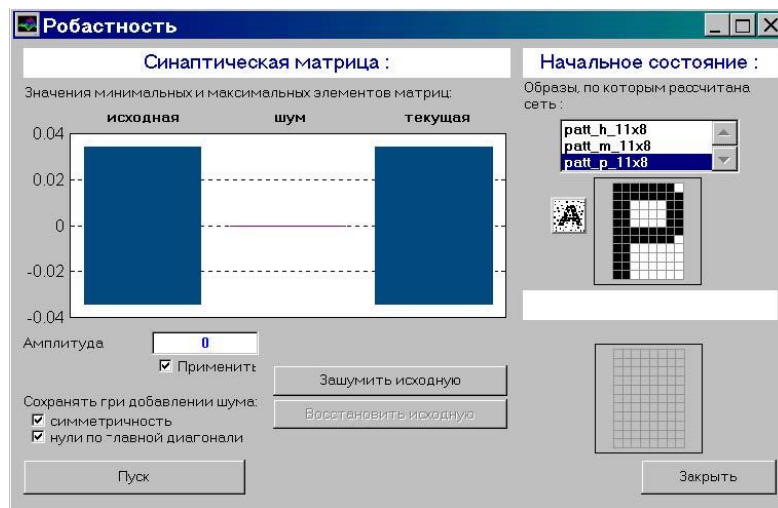


Рисунок 4.22 – Окно «Робастность»

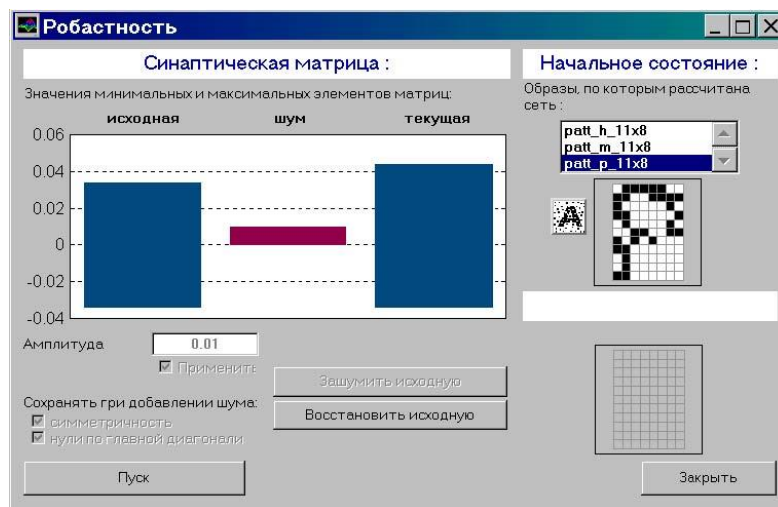


Рисунок 4.23 – Окно «Робастность». Начальные условия эксперимента

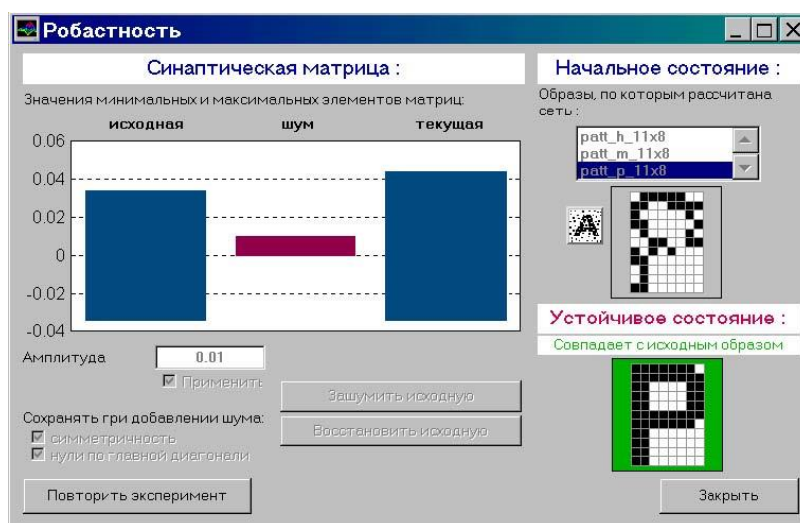


Рисунок 4.24 – Окно «Робастность». Правильное распознавание

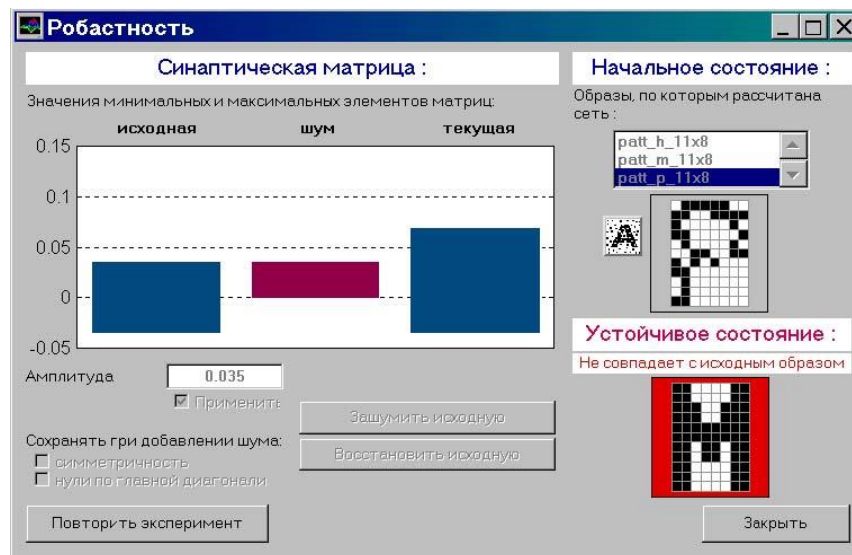


Рисунок 4.25 – Окно «Робастность». Неправильное распознавание

Содержание отчета

1. Название лабораторной работы, цель и программа работы.
2. Для каждой сети — список образов, на основании которых она рассчитана.
3. Для каждой сети – начальные условия и результаты всех экспериментов, указанных в п. «Порядок выполнения работы»
4. Анализ полученных результатов (ответить на вопрос, почему для четырех исследованных сетей в одних и тех же экспериментах получены разные результаты).

Контрольные вопросы к защите лабораторной работы

1. Что называется автоассоциативной памятью? Приведите пример.
2. Что называется аттрактором динамической системы? Объясните принцип применения динамических систем с множеством аттракторов для построения ассоциативной памяти.
3. Напишите уравнения динамики сети Хопфилда.
4. Какова активационная характеристика нейронов сети Хопфилда?
5. Чему равно начальное состояние нейронов сети Хопфилда?

6. Объясните различие синхронного и асинхронного режимов функционирования рекуррентной нейронной сети. Какой из режимов функционирования используется в сети Хопфилда?
7. Каким выражением определен энергетический функционал в процессе работы сети Хопфилда?
8. Почему время достижения сетью Хопфилда одного из аттракторов из произвольного начального состояния конечно?
9. Как рассчитывается матрица синаптических коэффициентов сети Хопфилда? Какими свойствами она обладает?
10. Как приближенно оценивается объем памяти сети Хопфилда?
11. На какие типы можно разделить множество аттракторов сети Хопфилда? Что такое “ложная память”?
12. Как применяется сеть Хопфилда для кластеризации данных? Как интерпретируются кластеры?
13. Как устанавливается класс принадлежности вектора признаков при использовании нескольких (по числу классов) сетей Хопфилда, “настроенных” на разные классы?
14. Как организован в лабораторной работе процесс оценивания размеров бассейнов аттракторов сети Хопфилда?

Список литературы

Перечень основной литературы:

1. Исаев, С. В. Интеллектуальные системы : учебное пособие / С. В. Исаев, О. С. Исаева. — Красноярск : Сибирский федеральный университет, 2017. — 120 с. — ISBN 978-5-7638-3781-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/84365.html> (дата обращения: 16.01.2024). — Режим доступа: для авторизир. пользователей.

2. Представление знаний в информационных системах : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, М. Ю. Серегин [и др.]. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2012. — 169 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/64163.html> (дата обращения: 16.01.2024). — Режим доступа: для авторизир. пользователей.

Перечень дополнительной литературы:

1. Загорулько, Ю. А. Инженерия знаний : учебное пособие / Ю. А. Загорулько, Г. Б. Загорулько. — Новосибирск : Новосибирский государственный университет, 2016. — 93 с. — ISBN 978-5-4437-0452-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/93454.html> (дата обращения: 24.07.2023). — Режим доступа: для авторизир. пользователей.

2. Цильковский, И. А. Методы анализа знаний и данных : конспект лекций / И. А. Цильковский, В. М. Волкова. — Новосибирск : Новосибирский государственный технический университет, 2010. — 68 с. — ISBN 978-57782-1377-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/45385.html> (дата обращения: 16.01.2024). — Режим доступа: для авторизир. пользователей.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение
высшего образования**
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
НЕВИННОМЫССКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)»**

ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ

Методические указания к практическим занятиям
для студентов направления подготовки
15.03.04 «Автоматизация технологических процессов и производств»

Невинномысск 2026

Методические указания предназначены для студентов направления подготовки 15.03.04 «Автоматизация технологических процессов и производств». Они содержат основы теории, порядок проведения практических занятий, перечень контрольных вопросов для самоподготовки и список рекомендуемой литературы.

Методические указания разработаны в соответствии с требованиями федерального государственного образовательного стандарта в части содержания и уровня подготовки выпускников по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств»

Составитель:

канд. техн. наук А.А. Евдокимов

Ответственный редактор:

канд. техн. наук Д.В. Болдырев

Содержание

Введение

Практическое занятие № 1 Аппроксимация функции правилами нечеткого вывода

Практическое занятие № 2 Исследование системы автоматического регулирования с нечетким регулятором

Практическое занятие № 3 Разработка нейро-нечеткой системы поддержки принятия решений при управлении технологическим процессом

Практическое занятие № 4 Разработка нейро-нечеткой системы поддержки принятия решений при диагностике работы подшипников качения

Практическое занятие № 5 Разработка нейро-нечеткой системы поддержки принятия решений при диагностике режимов работы турбогенератора

Список литературы

Введение

Проблемы создание эффективных систем управления на современном этапе связаны с повышением требований к качеству продукции и, как следствие этого, к качеству управления объектами в условиях недостаточной информации о закономерностях протекания управляемых процессов и неполноты (неточности) их математических моделей. Параметры функций управления определяются особенностями объекта. Инженерная и управленческая деятельность невозможна без развития у будущего специалиста инженерного профиля способностей к анализу информации, ее обработке и принятию собственных решений. Среди интеллектуальных управляющих систем особенно эффективны системы, построенные не на основе только одного из направлений искусственного интеллекта, а объединяющие в себе сразу несколько из них. В связи с этим в лабораторном практикуме уделено внимание решению задач на основе гибридной нейро-нечеткой технологии, где выводы делаются на основе аппарата нечёткой логики, но соответствующие функции принадлежности подстраиваются с использованием алгоритмов обучения нейронных сетей. Это позволит студентам уяснить, что сочетание нейронных сетей и нечёткой логики – принципиально различных конструкций – служит универсальным аппроксиматором сложных (нелинейных) функциональных зависимостей во многих интеллектуальных задачах.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1

АППРОКСИМАЦИЯ ФУНКЦИИ ПРАВИЛАМИ НЕЧЕТКОГО ВЫВОДА

Цель занятия: Изучение основных понятий теории нечетких множеств и их приложений.

Краткие теоретические сведения

Значительный шаг в направлении развития теории нечетких множеств сделал профессор Калифорнийского университета Лотфи А. Заде (1965 г. – публикация работы “Fuzzy Sets”). Заде расширил понятие множества, допустил, что характеристическая функция (функция принадлежности элемента множеству) может принимать любые значения в интервале $[0, 1]$. Такие множества были названы им нечеткими (fuzzy).

Пусть E – универсальное множество, x – элемент E , P – некоторое свойство. Обычное (четкое) множество A универсального множества E , элементы которого удовлетворяют свойству P , определяются как множество упорядоченных пар

$$A = \{ \mu_A(x) / x \},$$

где $\mu_A(x)$ – характеристическая функция, принимающая значение 1, если x удовлетворяет свойству P , и 0 – в противном случае.

В теории нечетких множеств для элементов x из E нет однозначного ответа «да/нет» относительно свойства P . В связи с этим нечеткое множество A универсального множества E определяется как множество упорядоченных пар с функцией принадлежности $\mu_A(x)$, принимающей значение в некотором упорядоченном множестве M (например, $M=[0, 1]$).

Функция принадлежности указывает степень (или уровень) принадлежности элемента x подмножеству A . Множество M называют множеством принадлежностей. Если $M=\{0, 1\}$, то нечеткое подмножество A может рассматриваться как обычное или четкое множество.

Примеры записи нечеткого множества.

1. Пусть $E = \{x_1, x_2, x_3, x_4, x_5\}$; A – нечеткое множество, для которого $\mu_A(x_1)=0,3$;
 $\mu_A(x_2)=0$; $\mu_A(x_3)=1$; $\mu_A(x_4)=0,6$; $\mu_A(x_5)=0,9$.

Тогда A можно представить в виде

$$A = \{0,3/x_1; 0/x_2; 1/x_3; 0,6/x_4; 0,9/x_5\}.$$

2. Пусть $E = \{1, 2, 3, \dots, 100\}$ и соответствует понятию «возраст», тогда нечеткое множество «молодой» может быть определено с помощью выражения

$$\mu_{\text{«молодой»}} = \begin{cases} 1, & x \in [1, 25], \\ \frac{1}{1 + \left(\frac{x - 25}{5}\right)^2}, & x > 25. \end{cases}$$

В приведенных выше примерах использованы прямые методы определения функций принадлежности, когда эксперт задает значение $\mu_A(x)$ для каждого $x \in E$. Такой способ используется для измеряемых понятий (скорость, температура и т.д.) или когда выделяют полярные значения. Разновидностью прямого способа задания функции принадлежности является групповой метод, когда группе экспертов предлагают сделать оценку того или иного явления, например, оценить: «этот человек лысый» или нет, – тогда количество утвердительных ответов, деленное на общее число экспертов, дает значение $\mu_{\text{лысый}}$ для данного лица.

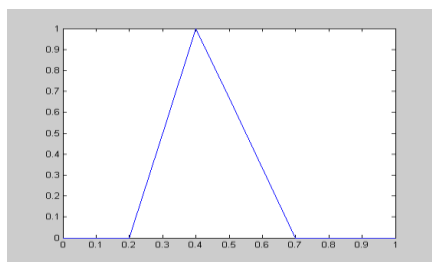
Кроме указанных способов задания функций принадлежности используют также типовые формы функций принадлежности (рисунок 1).

Аналитическая запись некоторых типовых функций принадлежности:

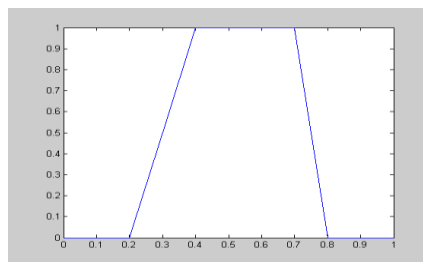
треугольная – $\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$, a, b, c – определяют параметры

функции; гауссова – $\mu(x) = e^{-\left(\frac{x-a}{b}\right)^2}$, a, b – параметры функции принадлежности; сигмоид-

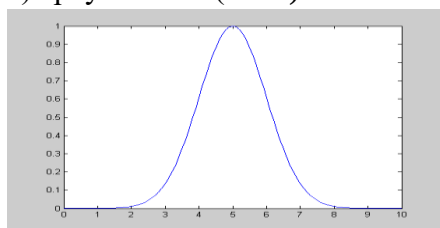
ная – $\mu(x) = \frac{1}{1 + \exp(-a(x-b))}$, a, b – параметры функции принадлежности.



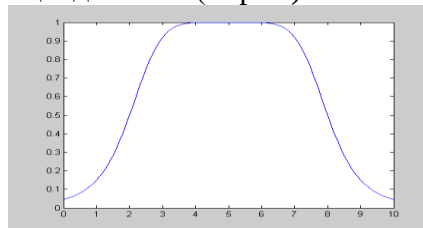
а) треугольная (trimf)



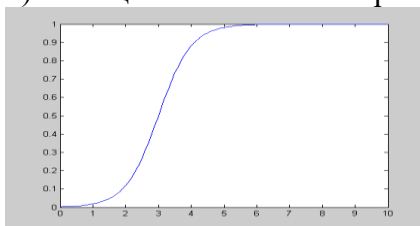
б) трапецеидальные (trapmf)



в) гауссова (gaussmf)



г) обобщенная колоколообразная (gbellmf)



д) сигмоидная (sigmf)

Рисунок 1 – Примеры типовых форм функций принадлежности (ниже в скобке приводится обозначение функции в среде MATLAB)

Основные операции над нечеткими множествами

Включение

Пусть A и B нечеткие множества на универсальном множестве E . A содержится в B , если $\forall x \in E \mu_A(x) \leq \mu_B(x)$. Обозначается $A \subset B$.

Равенство

A и B равны, если $\forall x \in E \mu_A(x) = \mu_B(x)$. Обозначение: $A=B$.

Дополнение

Пусть $M=[0,1]$, A и B дополняют друг друга, если $\forall x \in E \mu_A(x) = 1 - \mu_B(x)$. Обозначается $B = \bar{A}$ или $A = \bar{B}$.

Пересечение

$A \cap B$ – наименьшее нечеткое подмножество, содержащееся одновременно в A и B :
 $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \tilde{\wedge} \mu_B(x)$.

Или, по-другому, определение в классе треугольных норм (t – норма).

Типичными t – нормами являются [2, 3]:

- операция min как нечеткое логическое произведение с нечеткими переменными X_1, X_2 :

$$X_1 \tilde{\wedge} X_2 = \min(X_1, X_2);$$

- алгебраическое произведение $X_1 \tilde{\wedge} X_2 = X_1 \cdot X_2$.

Объединение

$A \cup B$ – наибольшее нечеткое подмножество, включающее как A , так и B , с функцией принадлежности:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \tilde{\vee} \mu_B(x).$$

Или, по другому, определение в классе треугольных конорм (s – норма).

Типичными s – нормами являются:

- операция нечеткого логического сложения $X_1 \tilde{\vee} X_2 = \max(X_1, X_2)$;

- алгебраическая сумма $X_1 \tilde{\vee} X_2 = X_1 + X_2 - X_1 X_2$,

где « $\sim$ » означает нечеткую логическую операцию.

Разность

$A - B = A \cap \bar{B}$ с функцией принадлежности: $\mu_{A-B}(x) = \min(\mu_A(x), 1 - \mu_B(x))$.

Нечеткие отношения

Нечеткое n-мерное отношение определяется как нечеткое подмножество R на E , принимающее свои значения в M . В случае $n=2$ и $M=[0,1]$ нечетким отношением R между множествами $X=E_1$ и $Y=E_2$ будет называться функция $R: (X, Y) \rightarrow [0,1]$, которая ставит в соответствие каждой паре элементов $(x, y) \in X \times Y$ величину $\mu_R(x, y) \in [0,1]$.

Обозначение: $x \in X, y \in Y : xRy$.

Алгебраические операции над нечеткими отношениями аналогичны операциям с нечетким множествам.

Композиция (свертка) двух нечетких отношений

Пусть R_1 – нечеткое отношение $R_1: (X \times Y) \rightarrow [0,1]$ между X и Y , и R_2 – нечеткое отношение $R_2: (Y \times Z) \rightarrow [0,1]$ между Y и Z . Нечеткое отношение между X и Z обозначается $R_1 \bullet R_2$ и определяется как

$$\mu_{R_1 \bullet R_2}(x, z) = \bigvee_y [\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)] = \max \left(\min(\mu_{R_1}(x, y), \mu_{R_2}(y, z)) \right),$$

где символ $\bigvee_y$ – обозначает операцию выбора наибольшего по y значения и называется

(max-min)-сверткой отношений R_1 и R_2 .

Пример. Пусть заданы отношения R_1 и R_2 .

Таблица 1

R_1	y_1	y_2	y_3
x_1	0,1	0,7	0,4
x_2	1	0,5	0

Таблица 2

R_2	z_1	z_2	z_3	z_4
y_1	0,9	0	1	0,2
y_2	0,3	0,6	0	0,9
y_3	0,1	1	0	0,5

$$\begin{aligned}\mu_{R_1 \bullet R_2}(x_1, y_1) &= [\mu_{R_1}(x_1, y_1) \wedge \mu_{R_2}(y_1, z_1)] \vee [\mu_{R_1}(x_1, y_2) \wedge \mu_{R_2}(y_2, z_1)] \vee \\ &\vee [\mu_{R_1}(x_1, y_3) \wedge \mu_{R_2}(y_3, z_1)] = \\ &= (0,1 \wedge 0,9) \vee (0,7 \wedge 0,3) \vee (0,4 \wedge 0,1) = 0,1 \vee 0,3 \vee 0,1 = 0,3.\end{aligned}$$

Таблица 3

$R_1 \cdot R_2$	z_1	z_2	z_3	z_4
x_1	0,3	0,6	0,1	
x_2	0,9	0,5	1	

Нечеткие выводы

В экспертных и управляющих системах механизм нечетких выводов в своей основе имеет базу знаний, формируемую специалистами предметной области в виде совокупности нечетких предикатных правил вида:

П₁: если x есть A_1 , то y есть B_1 ,

П₂: если x есть A_2 , то y есть B_2 ,

...

П_n: если x есть A_n , то y есть B_n ,

где x – входная переменная, y – переменная вывода, A и B – функции принадлежности, определенные на x и y соответственно.

Знания эксперта $A \rightarrow B$ отражает нечеткое причинное отношение предпосылки и заключения, поэтому его называют нечетким отношением:

$R = A \rightarrow B$,

где « $\rightarrow$ » – нечеткая импликация.

Отношение R можно рассматривать как нечеткое подмножество прямого произведения $X \times Y$ полного множества предпосылок X и заключений Y . Таким образом, процесс получения (нечеткого) результата вывода B' с использованием данного наблюдения A' и значения $A \rightarrow B$ можно представить в виде

$B' = A' \bullet R = A' \bullet (A \rightarrow B)$.

Алгоритм нечеткого вывода

1 **Нечеткость** (фаззификация, fuzzification). Функции принадлежности, определенные для входных переменных, применяются к их фактическим значениям для определения степени истинности каждой предпосылки каждого правила).

2 **Логический вывод**. Вычисленное значение истинности для предпосылок каждого правила применяется к заключениям каждого правила. Это приводит к одному нечеткому подмножеству, которое будет назначено переменной вывода для каждого правила. В качестве правил логического вывода используются только операции min (минимума) или prod (умножение).

3 **Композиция**. Нечеткие подмножества, назначенные для каждой переменной вывода (во всех правилах), объединяются вместе, чтобы сформировать одно нечеткое подмножество для каждой переменной вывода. При подобном объединении обычно используются операции max (максимум) или sum (сумма).

4 **Дефаззификация** – приведение к четкости (defuzzification). Преобразование нечеткого набора выводов в число.

Алгоритмы нечеткого вывода Мамдани (Mamdani)

Пусть заданы два нечетких правила:

П₁: если x есть A_1 и y есть B_1 , тогда z есть C_1 ,

П₂: если x есть A_2 и y есть B_2 , тогда z есть C_2 .

1. **Нечеткость**. Находят степени принадлежности для предпосылок каждого правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(y_0)$, $B_2(y_0)$.

2. **Нечеткий вывод**. Определяют уровни «отсечения» для предпосылок каждого правила (операция min):

$\alpha_1 = A_1(x_0) \wedge B_1(y_0)$,

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

определяют усеченные функции принадлежности

$$C'_1 = (\alpha_1 \wedge C_1(z)),$$

$$C'_2 = (\alpha_2 \wedge C_2(z)).$$

3. **Композиция.** Производится объединение найденных усеченных функций (операция max), получают нечеткое подмножество для переменной выхода с функцией принадлежности:

$$\mu_{\Sigma}(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

4. **Дефаззификация.** Приведение к четкости (определение z_0), например, *центроидным* методом (как центр тяжести для кривой $\mu_{\Sigma}(z)$):

$$z_0 = \frac{\int_{\Omega} z \mu_{\Sigma}(z) dz}{\int_{\Omega} \mu_{\Sigma}(z) dz}.$$

Алгоритм иллюстрируется на рисунке 2.

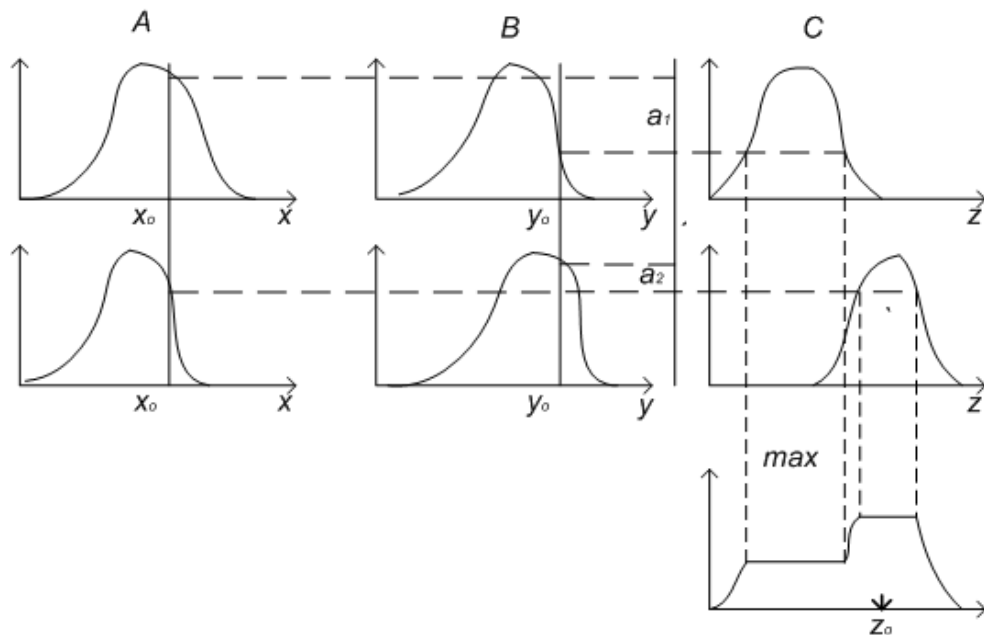


Рисунок 2 – Графическая интерпретация алгоритма Мамдани

Алгоритмы нечеткого вывода Сугено (Sugeno) и Такаги (Takagi)

Сугено (Sugeno) и Такаги (Takagi) использовали набор правил в следующей форме:

П₁: если x есть A_1 и y есть B_1 , тогда $z = a_1x + b_1y$,

П₂: если x есть A_2 и y есть B_2 , тогда $z = a_2x + b_2y$.

1 Первый этап – аналогично алгоритму Мамдани.

2 Определение $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$ и индивидуальные выходы правил:

$$z_1^* = a_1x_0 + b_1y_0,$$

$$z_2^* = a_2x_0 + b_2y_0.$$

3 Определяется значение переменной вывода:

$$z_0 = \frac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}.$$

Представленная форма правил иллюстрирует алгоритм Сугено 1-го порядка. Если правила записаны в форме:

П₁: если x есть A_1 и y есть B_1 , тогда $z = c_1$,

П₂: если x есть A_2 и y есть B_2 , тогда $z = c_2$,

то задан алгоритм Сугено 0-го порядка.

Иллюстрация алгоритма Сугено 0-го порядка представлена на рисунке 3.

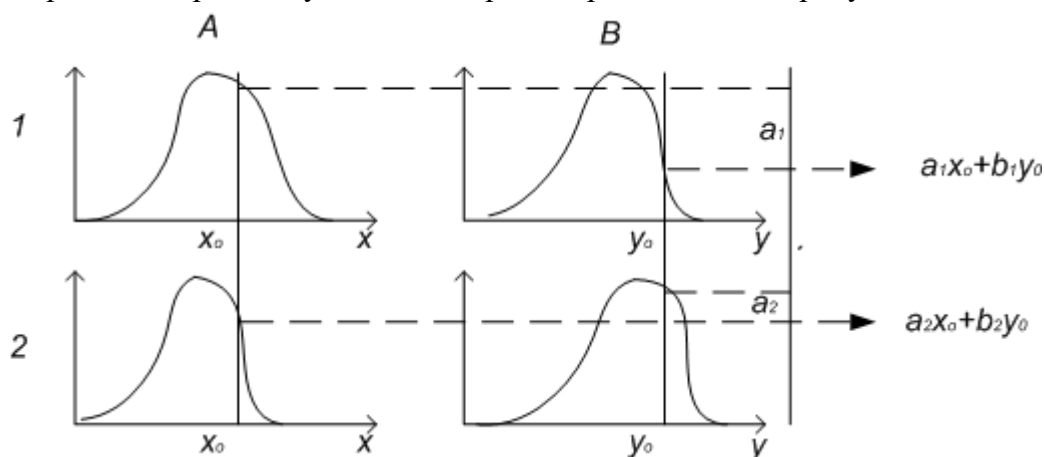


Рисунок 3 – Графическая интерпретация алгоритма Сугено

Перечислим наиболее известные методы дефаззификации.

Метод максимума – выбирается тот элемент нечеткого множества, который имеет наивысшую степень принадлежности этому множеству.

Если этот элемент не является единственным, т.е. функция принадлежности $\mu_A(y_1)$ имеет несколько локальных максимумов, или если имеется максимальное «плато», то выбор среди элементов, имеющих наивысшую степень принадлежности множеству, осуществляется на основе некоторого критерия.

Метод левого (правого) максимума – выбирается наименьшее (наибольшее) из чисел $y_1, y_2, \dots, y_n$, имеющих наивысшую степень принадлежности нечеткому множеству.

Метод среднего из максимумов – в качестве искомого четкого значения y_0 принимается среднее арифметическое координат локальных максимумов $y_0 = \frac{1}{n} \sum_{i=1}^n y_i$.

Возможные методы дефаззификации и их обозначения в MATLAB приведены на рисунке 4: наименьший максимум (smallest of max, som), наибольший максимум (largest of max, lom), средний максимум (mean of max, mom), бисекторный (bisector of area), рассмотренный выше центроидный (centroid).

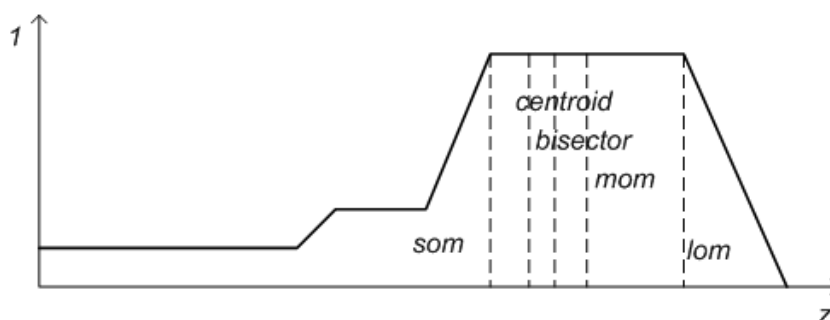


Рисунок 4 – Иллюстрация примеров фаззификации z

Пример проектирования системы нечеткого вывода в среде MATLAB

В качестве примера рассмотрим построение нечеткой аппроксимирующей функции $y=x^2$. Исходные данные для аппроксимации представлены в таблице 4.

Таблица 4 – Значения x и y

X	-1	-0,6	0	0,4	1
Y	1	0,36	0	0,16	1

1. Командой **fuzzy** из режима командной строки запускается основная интерфейсная программа пакета Fuzzy Logic – редактор нечеткой системы вывода.

2. В меню **File** выбрать команду New Sugeno FIS.
 3. Выбрать входной элемент системы *input1* и ввести в поле *Name* обозначение входной переменной (*x*).
 4. Войти в режим редактирования функции принадлежности – *Membership Function Editor* (двойное нажатие левой клавиши мыши). Выбором **Edit/Add MFs** (добавить функцию принадлежности) задать тип функции принадлежности (*gaussmf*) и количество (5).
 5. Установить диапазон (*Range*) изменения *x* от -1 до 1 (определяется заданным диапазоном *x*).
 6. Совместить ординаты максимумов функций принадлежности с значениями аргумента *x*.
Шаг выполняется двумя способами: выделить редактируемую функцию принадлежности, перетащить мышью кривую функции принадлежности; более точную установку проводят заданием числовых значений параметров функции принадлежности в поле *Params* (первое значение определяет размах кривой, второе – положение центра).
В поле *Name* вводится имя функции принадлежности (1- *bn*, 2 – *n*, 3 (центральная) – *z*, 4 – *p*, 5 – *bp*).
 - Выйти из редактора функций принадлежности – *Close*.
 7. Ввести в поле название выходной переменной *output1* (*y*), войти в режим редактирования функций принадлежности.
 8. Задать вид функции принадлежности для выходной переменной. Предлагается выбрать в качестве функции принадлежности линейные (*linear*) или постоянные (*constant*) – в зависимости от алгоритма Сугено (1-го или 0-го порядка). В поставленной задаче необходимо выбрать постоянные функции принадлежности с общим числом 4 (по числу различающихся значений *y*). Нажать **Ok**.
 9. Установить диапазон (*Range*) – [0, 1]. Изменить значения (*Params*) и задать для выходных переменных имена (*Name*) соответственно 0; 0,16; 0,36; 1. Закрыть редактор.
 10. Перейти в редактор правил (*Rule Editor*) (дважды щелкнуть на средний белый квадрат разрабатываемой структуры системы нечеткого вывода). При вводе каждого правила необходимо обозначить соответствие между каждой функцией принадлежности аргумента *x* и числовым значением *y*. Кривая, обозначенная *bn*, соответствует $y=1$, для чего выбирается в левом поле (с заголовком *x is*) вариант *bn*, а в правом – 1 и нажимается кнопка **Add rule**.
- Введенное правило появится в окне правил в виде
If (x is bn) then (y is 1) (1).
Аналогично вводятся все правила (всего 5).
Закрыть окно редактора правил и вернуться в окно FIS-редактора.
Построение системы закончено.

Сохранить на диске (**File/Save to disk as**) созданную систему. Перейти в редактор функций принадлежности (**View/Edit Membership function**). Из окна редактора командой можно перейти в окно просмотра правил (**View rules**), просмотра поверхности (**View surface**).

В окне просмотра правил иллюстрируется процесс принятия решения (вычисления *y*). Красная вертикальная черта, пересекающая графики в правой части окна, которую можно перемещать с помощью мыши, позволяет изменять значения переменной входа (либо вводят значение с клавиатуры в поле *Input*), при этом соответственно изменяется значение выхода. Просмотр кривой $y(x)$ осуществляется командой **View/View surface**. Большая погрешность аппроксимации заданной зависимости объясняется малым числом экспериментов. С помощью рассмотренных редакторов на любом этапе проектирования нечеткой модели можно внести необходимые коррективы, например задание пользовательской функции принадлежности.

В рассмотренном примере использованы следующие операции, устанавливаемые по умолчанию в алгоритме Сугено: логический вывод организуется с помощью операции логического умножения (*prod*); композиция организуется с помощью операции логической суммы (вероятностного ИЛИ, *probor*); приведение к четкости (дефазификация) организуется дискретным вариантом центроидного метода (взвешенным средним, *wtaver*).

Индивидуальное задание

Создать систему нечеткого вывода для аппроксимации функции $y=k \cdot x^2$ (вариант см. в таблице 6). Для аппроксимации взять 50 значений функции.

Таблица 6 – Варианты задания для аппроксимации функции

Вариант	1	2	3	4	5	6	7	8
k	1.5	1.8	1.3	1.9	2	2.2	1.4	2.5
mf input	trimf	gaussmf	trimf	gaussmf	trimf	trimf	trimf	gaussmf2

9	10	11	12	13	14	15	16	17	18
2.3	2.4	3.5	3.8	3.3	3.9	3	3.2	3.4	4
trimf	gaussmf	trimf	gaussmf	trimf	gaussmf	trimf	trimf	trimf	gaussmf

Контрольные вопросы

1. Что такое нечеткое множество?
2. Что характеризует функция принадлежности?
3. Какие методы дефазификации вам известны?
4. В чем суть нечеткого вывода по Мамдани?
5. В чем суть нечеткого вывода по Сугено
6. На чем основывается нечеткий вывод?

ИССЛЕДОВАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ С НЕЧЕТКИМ РЕГУЛЯТОРОМ

Цель занятия: Получить навык разработки нечеткого регулятора на базе продукционной модели представления знаний.

Краткие сведения из теории

Для создания одноконтурной системы управления воспользуемся программой Simulink пакета MATLAB. В Simulink необходимо собрать схему изображенную на рисунке 1.

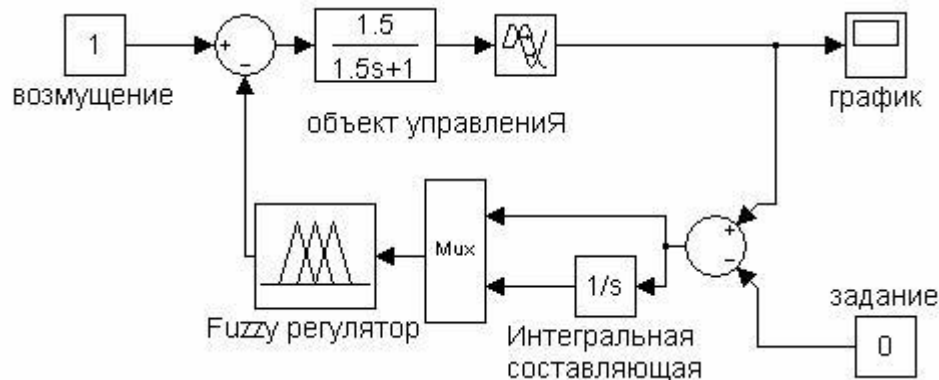


Рисунок 1 – Модель одноконтурной системы автоматического регулирования с ПИ-подобным fuzzy-регулятором

Теперь при помощи инструментов графического интерфейса пользователя (GUI) пакета "Fuzzy Logic Toolbox" создадим нечёткую систему, реализующую типовой аналоговый ПИ-регулятор. Заметим, что с помощью пакета "Fuzzy Logic Toolbox" можно строить нечеткие системы двух типов - Мамдани и Сугэно. Остановимся на системе типа Мамдани. Командой fuzzy в окне MATLAB вызываем окно Редактора фазы-инференционной системы (Fuzzy Inference System Editor), выбираем тип системы - Мамдани, задаём два входа – для пропорциональной и интегральной составляющих и называем входные переменные, например, x1 и x2, а выходную – y.

Из данного окна вызываем окно Редактора функций принадлежности (Membership Function Editor) двойным щелчком мыши по изображению переменной x1 или при помощи меню Edit. Здесь для лингвистического описания каждой переменной выберем семь треугольных термов (NB,NM,NS,ZE,PS,PM,PB). Термы выходной переменной лучше выбирать непересекающимися. Это повысит чёткость регулирования. В этом же окне зададим диапазоны изменения переменных:

Для входных переменных регулятора рекомендуются симметричные диапазоны изменения, при этом

$$x1 \in \left[-\frac{1}{k_1}; \frac{1}{k_1} \right] \text{ и } x2 \in \left[-\frac{1}{k_0}; \frac{1}{k_0} \right],$$

где K_1 , K_0 – оптимальные настройки пропорциональной и интегрирующей частей ПИ – регулятора в смысле какого-либо критерия.

Для выходной переменной регулятора диапазон изменения рекомендуется брать в виде $y \in [0; C]$, где верхняя граница C при единичном ступенчатом воздействии варьируется от 1.1 до 2, чтобы выходной сигнал регулятора мог компенсировать это возмущение. По мере увеличения значения C уменьшается динамическая ошибка, но возрастают время регулирования и число колебаний переходного процесса. Поэтому рекомендуется C принимать равным 2, когда наблюда-

ется оптимальное соотношение между величиной динамической ошибки, времени регулирования и количеством колебаний.

Теперь необходимо сформировать базу правил fuzzy-регулятора. В основу положен способ, предложенный в литературе. Линейный непрерывный ПИ-регулятор с дифференциальным уравнением

$$y(t) = k_D \cdot \varepsilon(t) + \frac{1}{T_E} \int_0^t \varepsilon(t) \cdot dt$$

можно заменить близким по стратегии и логике управления fuzzy-регулятором, если в качестве его выходной переменной рассматривать приращение управляющего воздействия Δy . Тогда ПИ закон регулирования можно представить в следующей дифференциальной форме:

$$\frac{dy(t)}{dt} = k_D \frac{d\varepsilon(t)}{dt} + \frac{1}{T_E} \varepsilon(t),$$

или в разностной форме:

$$\Delta y(k) = y(k) - y(k-1) = k_D \cdot \Delta \varepsilon(k) + \frac{\Delta t}{T_E} \cdot \varepsilon(k).$$

Таким образом, для входных переменных $\varepsilon(k)$ и $\Delta \varepsilon(k)$ и выходной $\Delta y(k)$ может быть синтезирован fuzzy-регулятор, реализующий нелинейный закон

$$\Delta y(k) = F[\Delta \varepsilon(k), \varepsilon(k)]$$

и эквивалентный в определенном смысле ПИ-регулятору.

Для нашего случая x_1 соответствует сигналу рассогласования $\varepsilon(k)$, x_2 соответствует приращению сигнала рассогласования $\Delta \varepsilon(k)$, а y соответствует $\Delta y(k)$. Лингвистические правила для такого ПИ-подобного fuzzy-регулятора приведены в таблице 1.

Вызываем окно Редактора правил (Rule Editor) в меню Edit или нажатием Ctrl+3 и заполняем список правилами из таблицы 1. Правила формируются по типу: ЕСЛИ ... И ..., ТО.... Можно посмотреть пространство управления, вызвав окно Просмотра пространства управления (Surface Viewer) из меню View или комбинацией клавиш Ctrl+6. Полученный файл сохраним под именем **fuzzy1.fis**.

В окне параметров блока Fuzzy Logic Controller укажем имя файла **fuzzy1**. В окне модели в меню **File** выберем пункт **Model Properties**. В открывшемся окне выберем вкладку **Callbacks** и в поле **Model pre-load function** напомним:

```
fuzzy1=readfis('fuzzy1').
```

Данная команда будет каждый раз при открытии файла модели помещать файл **fuzzy1.fis** в Workspace (рабочее пространство системы MATLAB). Это необходимо для нормального функционирования модели. Стоит заметить, что при внесении изменений в fis-файл нужно помещать его исправленную версию в Workspace либо при помощи пункта Export/To Workspace меню File, либо комбинацией клавиш Ctrl+T, либо каждый раз закрытием и открытием файла модели.

В диалоговом окне **Simulation Parameters** меню **Simulation** во вкладке **Advanced** для опции **Boolean logic signals** необходимо установить значение **off**. При этом блоки логики будут допускать переменные в форме с плавающей точкой.

Запуск модели – **Simulation/Start** (Ctrl+T).

В папке Demo находится файл с правилами нечеткого вывода. Загрузка файла осуществляется из окна редактора правил Rule Viewer: во вкладке **File/Import From Disk** открыть файл **fuzzy1.fsi**. Затем нужно поместить правила в Workspace системы (описано выше).

Работу разработанного fuzzy-регулятора можно сравнить с аналоговой системой регулирования (рисунок 2).

Дифференциальное уравнение ПИД-регулятора имеет вид:

$$y(t) = k_D \cdot \varepsilon(t) + \frac{1}{T_E} \int_0^t \varepsilon(t) \cdot dt + T_D \cdot \frac{d\varepsilon(t)}{dt}.$$

Таблица 1 – Лингвистические правила для ПИ fuzzy-регулятора

$\varepsilon \backslash \Delta\varepsilon$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Для исследования ПИ-регулятора в блоке настроек регулятора необходимо в поле параметра T_D (настройка дифференцирующей части ПИД-регулятора) установить значение 0.

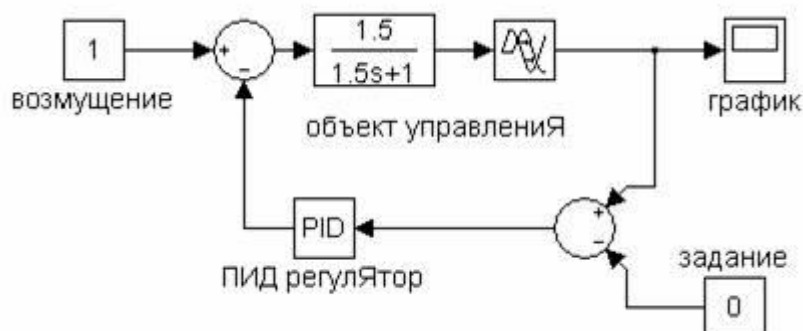


Рисунок 2 – Модель одноконтурной системы автоматического Регулирования

Индивидуальное задание

Исследовать АСР с аналоговым ПИ-регулятором и fuzzy-регулятором. В качестве модели объекта принять апериодическое звено, параметры объекта принять из таблицы 2.

Таблица 2 – Параметры объекта к заданию 5.2

Вариант	1	2	3	4	5	6	7	8
К	1.5	1.8	1.3	1.9	2	2.2	1.4	2.5
Т	1.8	1.8	1.5	1.5	1.3	1.5	1.5	1.6

9	10	11	12	13	14	15	16	17	18
2.3	2.4	3.5	3.8	3.3	3.9	3	3.2	3.4	4
1.4	1.7	1.8	1.5	1.5	1.5	1.8	1.2	1.4	1.9

Настройки ПИ-регулятора определить по формульному методу:

Регулятор	Апериодический процесс	Процесс с перерегулированием 20 %	Процесс с минимальным временем регулирования
ПИ	$K_1 = \frac{0,6 \cdot T}{K \cdot \tau},$ $K_0 = \frac{1}{K \cdot \tau}$	$K_1 = \frac{0,7 \cdot T}{K \cdot \tau},$ $K_0 = \frac{1}{K \cdot \tau}$	$K_1 = \frac{T}{K \cdot \tau},$ $K_0 = \frac{1}{K \cdot \tau}$

Контрольные вопросы

1. Какие критерии качества переходного процесса свидетельствуют о преимуществе классического регулятора, а какие – о преимуществе нечеткого регулятора?
2. Как формируется набор правил для нечеткого регулятора?
3. Есть ли преимущества в настройке нечеткого регулятора?
4. Изобразите структуру нечеткого регулятора.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

РАЗРАБОТКА НЕЙРО-НЕЧЕТКОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ УПРАВЛЕНИИ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ

Цель занятия – Ознакомление с методикой применения нейро-нечеткой системы поддержки принятия решений при управлении технологическим процессом.

1 Гибридные сети

Для подавляющего большинства как искусственных, так и естественных объектов управления (ОУ), которыми необходимо управлять, построение точных математических моделей практически невозможно ввиду их плохой формализуемости. К тому же эти объекты могут функционировать в среде, свойства которой изменяются или же вообще не могут быть определены заранее. Управление такими объектами возможно только с использованием адаптивных принципов. В случае плохой формализуемости ОУ особый интерес вызывают системы, построенные на новых, интеллектуальных принципах. Эти системы используют наработки таких направлений искусственного интеллекта (ИИ) как нечёткая логика, экспертные системы, искусственные нейронные сети и т. д.

Среди всех интеллектуальных управляющих систем особенно эффективны системы, построенные не на основе только одного из вышеперечисленных направлений ИИ, но объединяющие в себе сразу несколько из них.

В качестве примера рассмотрена система принятия решений при управлении технологическим процессом гидрирования бутанола, основанная на применении гибридной сети, где выводы делаются на основе аппарата нечёткой логики, но соответствующие функции принадлежности подстраиваются с использованием алгоритмов обучения нейронных сетей.

Нейронные сети и нечёткая логика – принципиально различные конструкции – являются универсальными аппроксиматорами сложных (нелинейных) функциональных зависимостей во многих интеллектуальных задачах.

Главной особенностью нейронных сетей является их способность к обучению. Она заключается в применении специально разработанных алгоритмов, среди которых наиболее популярно правило «обратного распространения ошибки» (back-propagation). Для обучения нейронной сети не требуется никакой априорной информации о структуре искомой функциональной зависимости. Необходима лишь обучающая выборка в виде экспериментальных пар «входы – выход». Однако обученная нейронная сеть – граф со взвешенными дугами – не поддаётся содержательной интерпретации.

Достоинством нечёткой логики является возможность использования экспертных знаний о структуре объекта в виде лингвистических высказываний: если <входы>, то <выход>.

Кроме того, нечёткие контроллеры могут работать с не полностью описанными системами с неизвестной динамикой, так как для них не требуется априорная математическая модель объекта управления.

Однако аппарат нечёткой логики не содержит механизмов обучения. Поэтому результаты нечёткого логического вывода существенно зависят от вида функций принадлежности, формализующих нечёткие термы: малый, большой, холодный, горячий и т. п.

Вообще говоря, теоретически, системы с нечёткой логикой и искусственные нейронные сети эквивалентны друг другу, однако на практике у них имеются собственные достоинства и недостатки. Данное соображение легло в основу аппарата гибридных сетей, где выводы делаются на основе аппарата нечёткой логики, но соответствующие функции принадлежности подстраиваются с использованием алгоритмов обучения нейронных сетей, например алгоритма обратного распространения ошибки. Такие системы не только используют априорную информацию, но могут приобретать новые знания и для пользователя являются логически прозрачными.

2 Описание технологического процесса

Узел гидрирования бутанола под давлением предназначен для дополнительного гидрирования карбонильных и непредельных соединений, входящих в состав обезвоженного бутанола, на хромоникелевом катализаторе. Структурная схема узла представлена на рисунке 1.

Обезвоженный бутанол из куба колонны поз. 71 поступает в емкость поз. Е-150^А, откуда насосом поз. Н-108/1,2, через подогреватель поз.

П-109 подается в верхнюю часть реактора поз. Р-110 или поз. Р-110-а.

Водород из цеха № 12-а подается в буферный сосуд поз. Е-114, с давлением не менее 16,0 кгс/см² по линии № 828. Из буферного сосуда (буфера) поз. Е-114 водород поступает в подогреватель поз. П-141, нагревается до температуры 120-150°С перегретым паром давлением Р-6,0 кгс/см² и далее подается в реактор поз. Р-110 или поз. Р-110-а, в верхнюю часть

Гидрирование карбонильных и непредельных соединений в реакторе поз. Р-110 или поз. Р-110-а протекает на хромоникелевом катализаторе в жидкой фазе при температурах в зонах реакции 115-155°С и давлении 7,0-15 кгс/см².

Из реактора поз. Р-110, Р-110-а смесь бутанола и водорода поступает в сепаратор поз. С-111, где происходит отделение бутанола от водорода. Бутанол из сепаратора поз. С-111 поступает в сепаратор поз. С-142, где, вследствие понижения давления до Р – 6 кгс/см², происходит выделение из бутанола растворённого в нём водорода.

Освободившийся от водорода бутанол из сепаратора поз. С-142 подается на 6-ю или 11-ю тарелку колонны ректификации поз. 75. Технологической схемой предусмотрена возможность выдачи бутанола из сепаратора поз. С-111 в колонну поз. 75 минуя сепаратор поз. С-142.

Водород из сепаратора поз. С-111 поступает в холодильник поз. Х-117 или, минуя его, в сепаратор поз. С-118. В холодильнике поз. Х-117 водород охлаждается захлажденной водой и поступает в сепаратор поз. С-118.

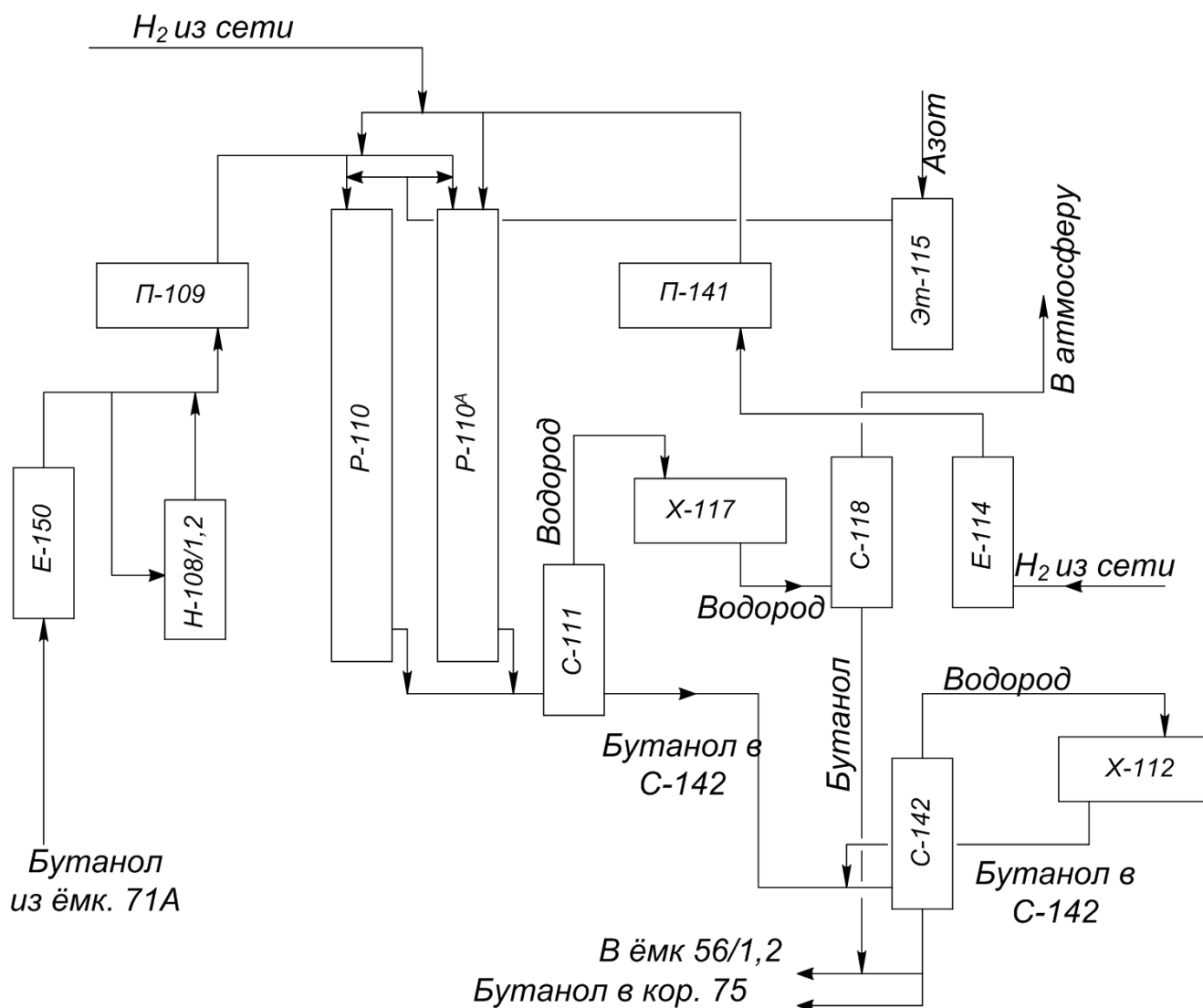


Рисунок 1 – Структурная схема узла гидрирования под давлением

В сепараторе поз. С-118 происходит дополнительное отделение бутанола от водорода. Водород, освободившийся от бутанола, из сепаратора поз. С-118 направляется на всас инжектора поз. 64/1-5.

Водород из сепаратора поз. С-142 направляется в холодильник поз. Х-112, где охлаждается захоложенной водой.

Сконденсировавшиеся пары бутанола, унесённые из сепаратора поз. С-142, сливаются из холодильника поз. Х-112 в сепаратор поз. С-142. Освободившийся от бутанола водород направляется на всас инжектора поз. 64/1-5.

По мере накопления, бутанол из сепаратора поз. С-118 сливается в емкость поз. 56/1,2. Технологической схемой предусмотрена возможность слива бутанола из сепаратора поз. С-142 в

емкость поз. 56/1,2, а также возможность сброса водорода из сепараторов поз. С-118 и поз. С-142 в атмосферу через выхлопной стояк.

Загрузка, сушка и восстановление хромоникелевого катализатора.

Перед загрузкой в реактор поз. Р-110, Р-110-а катализатор просеивается от пыли через сито.

В каждую секцию реактора поз. Р-110, Р-110-а загружается по 0,5 м³ катализатора на высоту 1650 мм (3 секции).

С целью удаления микроскопической влаги из катализатора, после загрузки в реактор поз. Р-110, Р-110-а, производится его сушка азотом при температуре не более 160°С под давлением до 2 кгс/см².

Азот на сушку катализатора в реактор поз. Р-110, Р-110-а подается через подогреватель поз. П-141, в котором подогревается паром давлением 6 кгс/см² до температуры 130-140°С, и дополнительно – в электроподогревателе поз. Эт-115 до температуры 200°С.

Подача пара в подогреватель поз. П-141 регулируется так, чтобы скорость подъема температуры азота на выходе из подогревателя не превышала 10-15°С в час.

Азот из нижней части реактора поз. Р-110, Р-110-а через гидрозатвор поз. 67-а сбрасывается в атмосферу.

По прекращении выделения воды катализатор подвергается восстановлению водородом при температуре не более 250°С и давлении до 3 кгс/см².

Азотоводородная смесь на восстановление катализатора в реактор поз. Р-110, Р-110-а подается из буфера поз. Е-114 через подогреватель поз. П-141 и электроподогреватель поз. Эт-115.

Из нижней части реактора поз. Р-110, Р-110-а азотоводородная смесь сбрасывается в атмосферу через гидрозатвор поз. 67-а.

Скорость поглощения водорода устанавливается не более 30 м³/ч, а скорость подъема температуры азотоводородной смеси не более 10-15°С в час.

В случае завышения температуры катализатора, схемой предусмотрена возможность подачи холодного водорода из сети в верхнюю часть реактора поз. Р-110, Р-110-а.

Процесс восстановления катализатора продолжается до прекращения выделения воды из азотоводородной смеси и до отсутствия разницы концентраций водорода до и после реактора поз. Р-110, Р-110-а.

Выгрузка отработанного хромоникелевого катализатора. Отработанный хромоникелевый катализатор, перед выгрузкой из реактора поз. Р-110, Р-110-а, охлаждается азотом до температуры 40-50°С, смачивается водой и, через выгрузочные люки, вручную, выгружается в бочки.

Нормы технологического процесса приведены в таблице 1.

Таблица 1 – Нормы технологического процесса

Наименование показателя	Нормы
1. Уровень бутанола в ёмкости Е-150	20-80 % шкалы

	прибора
2. Температура бутанола на выходе из подогревателя П-109	130-150°C
3. Давление водорода в буфере Е-114	н/б 27 кгс/см ²
4. Расход водорода в реактор Р-110, 110А: - при работе - при восстановлении катализатора	н/б 900 нм ³ /ч н/б 3000 нм ³ /ч
5. Температура водорода на входе в реактор Р-110, 110А: - при работе - при восстановлении катализатора	120-150°C н/б 300°C
6. Расход бутанола на реактор Р-110, 110А	н/б 7,5 т/ч
7. Перепад давления “низ-верх” в реакторе Р-110, 110А	н/б 1,0 кгс/см ²
8. Температура в реакторе Р-110, 110А (верх, середина, низ)	135-155 °C
9. Уровень в сепараторе С-111	н/б 80 % шкалы прибора
10. Уровень в сепараторе С-118	н/б 80 % шкалы прибора
11. Давление в сепараторе С-118	7-15 кгс/см ²
12. Уровень в сепараторе С-142	30-80 % шкалы прибора
13. Давление в сепараторе 142	н/б 6 кгс/см ²
14. Температура на выходе из электроподогревателя Э-115	н/б 300 °C

2 Составление обучающей выборки

Первым этапом решения данной задачи является правильное задание обучающей выборки. Основой стабильного функционирования технологического процесса является соблюдение технологического регламента. В постоянном технологическом регламенте производства бутилового спирта приведён список возможных отклонений от нормального технологического режима и методов их устранения (таблица 2).

Технологический процесс гидрирования бутанола характеризуется двумя режимами: первый режим – режим восстановления катализатора, второй – рабочий режим. Диапазоны значений контролируемых параметров в каждом из режимов значительно различаются, поэтому эти режимы следует рассмотреть в отдельности. Кроме того, разделение на два режима позволит значительно упростить вычисления.

В соответствии с регламентом можно сформировать две обучающие выборки, соответствующие каждому из режимов.

Исследуемую установку можно рассмотреть как объект вида

$$y = f(x_1, x_2, \dots, x_n), \quad (1)$$

для которого связь «входы x_i – выход y » можно представить в виде экспертной матрицы знаний. В таблицах 3 и 4 приведены матрицы знаний для режима восстановления катализатора и рабочего режима соответственно. Этим матрицам соответствует нечёткая база знаний:

ЕСЛИ $[(x_1=a_1^{j1}) \text{ И } \dots (x_i=a_i^{j1}) \text{ И } \dots (x_n=a_n^{j1})]$ (с весом w_{j1}) ...
... ИЛИ $[(x_1=a_1^{jkj}) \text{ И } \dots (x_i=a_i^{jkj}) \text{ И } \dots (x_n=a_n^{jkj})]$ (с весом w_{jk}),

ТО $y = d_j, \quad j = 1, m,$

где a_i^p – лингвистический терм, оценивающий переменную x_i в строке $p=k_j$; k_j – количество строк-конъюнкций, соответствующих классу d_j выходной переменной y ; w_{jp} – число в диапазоне $[0, 1]$, характеризующее субъективную меру уверенности эксперта относительно высказывания с номером $p = k_j$ [11].

Следует отметить, что при составлении обучающей выборки особое внимание уделялось зонам перехода значений контролируемых параметров за предельные значения, так как эти зоны по сути являются зонами принятия решения.

В таблице 3 значение графы <выход> «0» означает нормальное течение технологического процесса; значение «1» – увеличение температуры в реакторе при восстановлении катализатора более 300°C; значение «2» – высокая температура водорода, идущего на реактор, свыше 300°C; значение «3» – прекращение подачи водорода или снижение его давления менее 19 кгс/см².

В таблице 4 значение графы <выход> «0» означает нормальное течение технологического процесса; значение «1» – в реакторе резко растёт температура в слое катализатора; значение «2» – прекращение подачи водорода или снижение его давления менее 19 кгс/см²; значение «3» – увеличивается перепад давления в реакторе более 1,0 кгс/см².

Таблица 2 – Возможные отклонения от нормального технологического режима и методы их устранения

Отклонение от нормального технологического режима	Возможные причины отклонения	Методы устранения отклонения и действия персонала
1. Увеличилась скорость поглощения водорода при восстановлении более 30,0 м ³ /час	Высокая температура водорода, идущего на реактор, свыше 300° С	Снизить температуру водорода для чего снизить напряжение на нагрев спирали аппарата поз. 115

2. Увеличилась температура в реакторе при восстановлении катализатора более чем на выходе из аппаратов поз. 115, 141	Происходит самовозгорание катализатора	1. Прекратить подачу водорода. 2. Подать на установку азот. 3. Снизить напряжение на нагрев спирали аппарата поз. 115
3. Прекращение подачи водорода или снижение его давления менее 19 кгс/см ²	Неполадки в цехе № 1-В или в цехе № 12-А	1. Прекратить подачу бутанола на гидрирование и остановить установку 2. Установку поставить под давление азота
4. Увеличивается перепад давления в реакторе более 1,0 кгс/см ²	Разрушился катализатор	Заменить катализатор
5. При подаче бутанола в реактор резко растёт температура в слое катализатора	1. Высокая активность катализатора. 2. Большая подача бутанола	1. Прекратить подачу бутанола, дать в реактор холодный водород 2. После снижения температуры бутанол подавать в минимальных количествах

Таблица 3 – Экспертная матрица знаний для режима восстановления катализатора

Номер правила	ЕСЛИ <входы>			ТО <выход>	Вес правила
	Температура в реакторе, °С	Температура на входе в реактор, °С	Давление в бункере поз. 114, кгс/см ²		
1	299	299	19	0	1
2	299	299	24	0	1
3	299	270	22	0	1
4	299	270	25	0	1
5	270	290	25	0	1
6	270	270	25	0	1
7	315	299	19	1	1
8	315	270	19	1	1
9	315	270	25	1	1
10	301	299	19	1	1
11	301	299	25	1	1
12	301	270	19	1	1
13	301	270	25	1	1
14	299	315	19	2	1
15	270	315	19	2	1
16	270	315	25	2	1

17	299	301	19	2	1
18	299	301	25	2	1
19	270	301	19	2	1
20	270	301	25	2	1
21	299	299	17,5	3	1
22	299	299	18,9	3	1
23	270	299	17	3	1
24	270	299	18,9	3	1
25	270	270	18,5	3	1
26	270	270	18,9	3	1
27	299	270	17	3	1

Таблица 4 – Экспертная матрица знаний для рабочего режима

Номер правила	ЕСЛИ <входы>			ТО <выход>	Вес правила
	Температура в реакторе, °С	Давление в буфере поз. 114, кгс/см ²	Перепад давления в реакторе, кгс/см ²		
1	150	19	0,95	0	1
2	120	19	0,95	0	1
3	150	25	0,95	0	1
4	150	25	0,1	0	1
5	120	19	0,1	0	1
6	150	25	0,1	0	1
7	151	19	0,95	1	1
8	160	25	0,95	1	1
9	155	19	0,95	1	1
10	151	19	0,1	1	1
11	159	19	0,1	1	1
12	160	25	0,1	1	1
13	160	19	0,95	1	1
14	150	18,9	0,95	2	1
15	120	18,9	0,95	2	1
16	150	15	0,95	2	1
17	120	15	0,1	2	1
18	150	18,9	0,1	2	1
19	120	15	0,1	2	1
20	150	15	0,1	2	1
21	150	19	1,05	3	1
22	120	19	1,05	3	1
23	150	25	1,5	3	1
24	120	25	1,5	3	1
25	120	19	1,5	3	1
26	150	25	1,05	3	1
27	120	25	1,05	3	1

3 Определение структуры нейро-нечёткой сети

Лингвистическую информацию об объекте можно представить в виде специальной нейро-нечёткой сети. Структура такой сети представлена на рисунке 2, а функциональные особенности узлов показаны в таблице 5.

Из рисунка 2 видно, что нейронечёткая сеть имеет пять слоёв: 1 – входы объекта идентификации; 2 – нечёткие термы, используемые в базе знаний; 3 – строки-конъюнкции нечёткой базы знаний; 4 – правила, объединяемые в классы; 5 – операция дефазификации, т. е. преобразование результатов нечёткого логического вывода в чёткое число.

Число узлов в нейронечёткой сети определяется так: слой 1 – по количеству входов объекта идентификации; 2 – по количеству нечётких термов в базе знаний; слой 3 – по количеству строк-конъюнкций в базе знаний; слой 4 – по количеству классов, на которые разбивается диапазон выходной переменной.

Дугам графа присваиваются следующие веса: единица (дуги между первым и вторым слоями); функции принадлежности входа к нечёткому терму (дуги между вторым и третьим слоями); веса правил (дуги между третьим и четвёртым слоями); единица (дуги между четвёртым и пятым слоями).

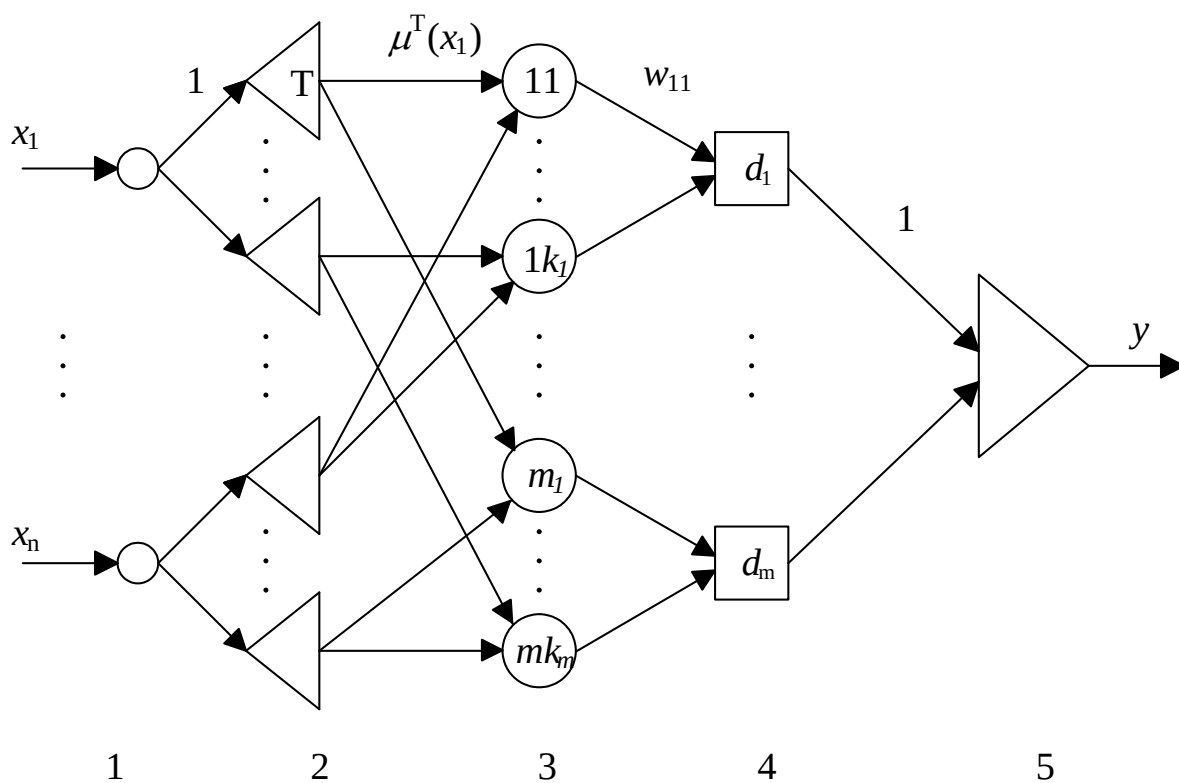
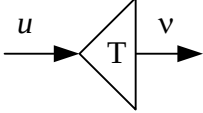
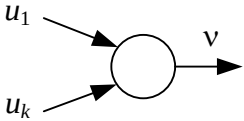
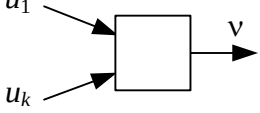
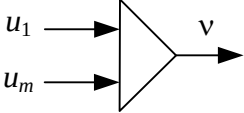


Рисунок 2 – Структура нейро-нечёткой сети

Таблица 5 – Элементы нейро-нечеткой сети

Узел сети	Название узла	Функция узла
$u \rightarrow \bigcirc \rightarrow v$	Вход	$v = u$

	Нечёткий терм	$v = \mu^T(u)$
	Нечёткое правило	$v = \prod_{i=1}^k u_i$
	Класс правил	$v = \sum_{i=1}^k u_i$
	Дефазификация	$v = \sum_{j=1}^m u_j d_j / \sum_{j=1}^m u_j$

4 Обучение нейро-нечёткой сети

Суть обучения состоит в подборе таких весов дуг, которые минимизируют различие между результатами нейро-нечёткой аппроксимации и реальным поведением объекта. Для обучения используется система рекуррентных соотношений

$$w_{jp}(t+1) = w_{jp}(t) - \eta \cdot \frac{\partial E_t}{\partial w_{jp}(t)}, \quad (2)$$

$$c_i^{jp}(t+1) = c_i^{jp}(t) - \eta \cdot \frac{\partial E_t}{\partial c_i^{jp}(t)}, \quad (3)$$

$$b_i^{jp}(t+1) = b_i^{jp}(t) - \eta \frac{\partial E_t}{\partial b_i^{jp}(t)}, \quad j=1, m, \quad i=1, n, \quad p=k_j, \quad (4)$$

минимизирующих критерий

$$E_t = \frac{1}{2}(\hat{y}_t - y_t)^2, \quad (5)$$

который применяется в теории нейронных сетей, где $\hat{y}_t$ и y_t – теоретический и экспериментальный выходы объекта на t-м шаге обучения; $w_{jp}(t)$, $c_{jp}(t)$, $b_{jp}(t)$ – веса правил w и параметры функций принадлежности b, c на t-м шаге обучения; η – параметр обучения, который может выбираться в соответствии с рекомендациями работы.

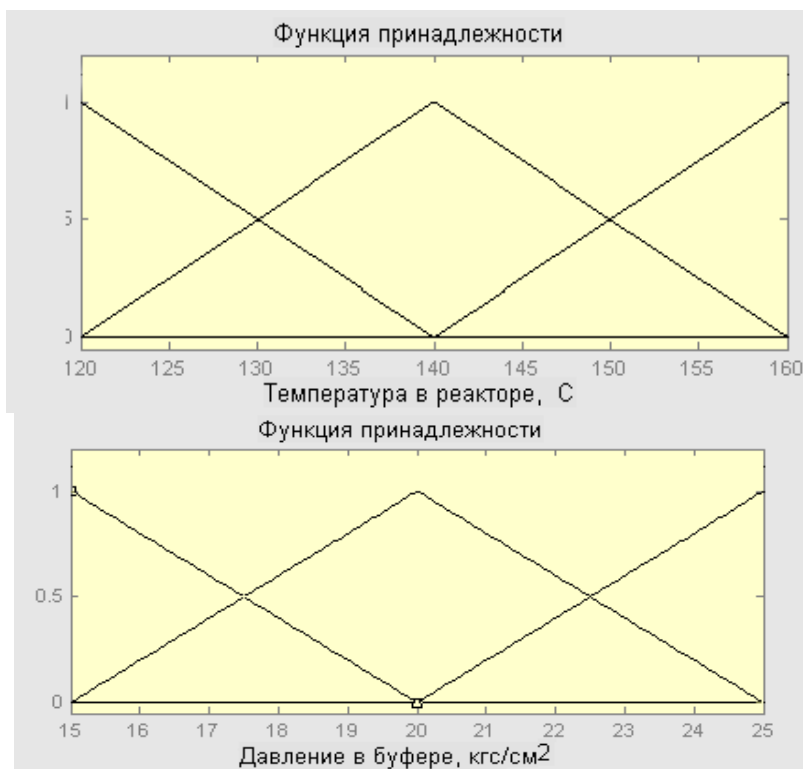
Функции принадлежности нечётких термов, используемых в базе знаний, выбираются экспертно.

Алгоритм обучения нейро-нечёткой сети включает два этапа. На первом этапе вычисляется модельное значение выхода объекта y , соответствующее заданной архитектуре сети. На втором этапе вычисляется значение невязки E_t и пересчитываются веса межнейронных связей.

4.1 Подбор функций принадлежности

Принадлежность каждого точного значения к одному из термов лингвистической переменной определяется посредством функций принадлежности (ФП). Её вид может быть абсолютно произвольным. Наиболее предпочтительными формами ФП являются треугольная и трапециидальная, поскольку нечёткий вывод значительно упрощается при использовании кусочно-линейных ФП.

Исследуемая установка гидрирования бутанола, представленная в виде нелинейного объекта, была задана в виде аналитической зависимости, т. е. в виде обучающих выборок, соответствующих данным таблиц 3 и 4. Для режима восстановления катализатора для каждого технологического параметра были выбраны трапециидальные функции принадлежности, а для рабочего режима выбраны треугольные функции принадлежности (рисунок 3).



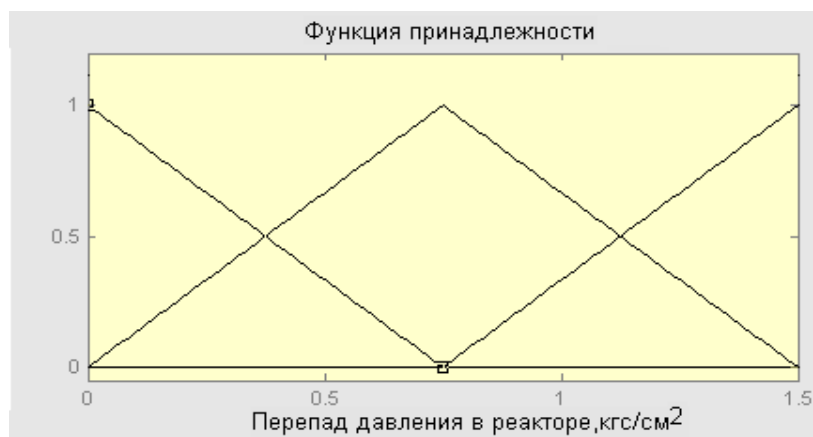


Рисунок 3 – Функции принадлежности входных параметров до обучений нейро-нечёткой сети

Посредством этих ФП входные величины фазифицируются, т. е. переходят из чёткого значения в принадлежность нечёткого множества. Далее необходимо выбрать алгоритм нечёткого вывода. Алгоритмы нечёткого вывода различаются главным образом видом нечёткого вывода, следующим после фазификации, и разновидностью метода дефазификации. Наиболее распространёнными алгоритмами являются: алгоритм Мамдани, алгоритм Сугено, алгоритм Ларсена, алгоритм Цукамото.

Результаты нечёткого вывода, полученные разными моделями, могут различаться. Результаты сравнения данных алгоритмов применительно к задачам управления динамическими объектами говорят в пользу алгоритма Сугено, вследствие его меньшей вычислительной сложности, по сравнению с другими алгоритмами.

Исходя из вышесказанного, применим алгоритм нечёткого вывода Сугено. Итоговая величина определяется как средневзвешенное.

4.2 Моделирование системы принятия решения

Для моделирования гибридной сети используем пакет прикладных программ MATLAB Fuzzy Logic Toolbox. Данный пакет обладает широкими возможностями для аппроксимации нелинейных зависимостей адаптивными нечёткими моделями. Основные функции и алгоритмы в расширении Fuzzy Logic Toolbox реализованы для механизма вывода Сугено. На первом этапе проводится генерация правил и нахождение границ термов. На втором этапе используется технология ANFIS (Adaptive Network-based Fuzzy Inference System) – итерационная процедура для настройки функций принадлежности методом обратного распространения ошибки. Адаптивная нечёткая модель способна настроиться под конкретный способ дефазификации.

Используя приложение пакета MATLAB ANFISedit генерируем данные, соответствующие выборкам таблиц 3 и 4.

Системой MATLAB в соответствии с введёнными условиями и параметрами сформирована нейронечёткая сеть, структура которой представлена на рисунке 4. Из структуры сети видно, что она имеет три входа, три внутренних слоя и один выход.

Для рабочего режима первый вход – значение температуры в реакторе, которое должно поддерживаться в пределах $120^{\circ}\text{--}150^{\circ}\text{C}$; второй вход – значение давления водорода в буферной ёмкости поз. 114, которое для стабильности протекания процесса гидрирования бутанола должно поддерживаться в диапазоне $19\text{--}25\text{ кгс/см}^2$; третий вход – значение перепада давления «верх – низ» в реакторе, с помощью которого отслеживают состояние катализатора, и которое не должно превышать более $1,0\text{ кгс/см}^2$.

Второй слой сети состоит из девяти нечётких термов по три на каждый из входов.

Третий слой сети состоит из двадцати семи элементов (нечётких правил), количество которых соответствует количеству строк-конъюнкций, приведённых в базе знаний (таблица 4).

Последний слой выполняет операцию дефазификации, т. е. преобразует результат нечёткого логического вывода, сделанного нейронечёткой сетью, в чёткое число, которое должно соответствовать номеру инструкции, которой необходимо руководствоваться в той или иной сложившейся ситуации.

Дугам графа нейронечёткой сети присвоены следующие веса:

- 1) единица – между первым (входным) и вторым слоем;
- 2) функции принадлежности входа к нечёткому терму – между вторым и третьим слоями.

Дугам графа сети для пускового режима присвоены трапециидальные функции принадлежности, а дугам графа сети для рабочего режима присвоены треугольные функции принадлежности;

3) между третьим и четвёртым слоями – веса правил. Основным руководством при составлении экспертной матрицы знаний являлся «Постоянный технологический регламент производства н-бутилового спирта цеха № 9 ОАО «Невинномысский Азот», поэтому всем весам правил (субъективным мерам уверенности эксперта относительно высказываний) присвоена единица.

- 4) единица – между четвёртым и пятым слоями.

В качестве метода настройки параметров был задан метод «обратного распространения ошибки» (back-propagation), уровень ошибки «0», а количество циклов обучения – 3000.

Результат моделирования системы в виде графика ошибки сети в зависимости от числа проведённых циклов обучения представлен на рисунок 5. График обучения нейронечёткой сети показал, что число итераций обучения для трёх входных параметров составляет 2900, а время обучения – 4 мин. Ошибка обучения составила 0,00179 или 0,179 %.

Полученные результаты обучения нейронечёткой сети для рабочего режима установки гидрирования бутанола представлены в виде диаграмм (рисунки 6–7).

На диаграммах входу по номером «1» соответствует значение температуры внутри реактора; входу под номером «2» – значение давления водорода в буферной ёмкости; входу под номером

«3» – значение перепада давления в реакторе; выход – определяет течение технологического процесса.

На рисунке 6 представлены диаграммы, соответствующие нормальному течению технологического процесса. Значение температуры в реакторе задано 140°C (вход 1), давление водорода в буфере равно 23 кгс/см^2 (вход 2), перепад давления «верх-низ» в реакторе – $0,75 \text{ кгс/см}^2$ (вход 3). Выходное значение равно 0,00257 – это означает, что технологический процесс проходит без отклонений. Выходное значение близкое к нулю получается и при варьировании входных значений в пределах допустимых.

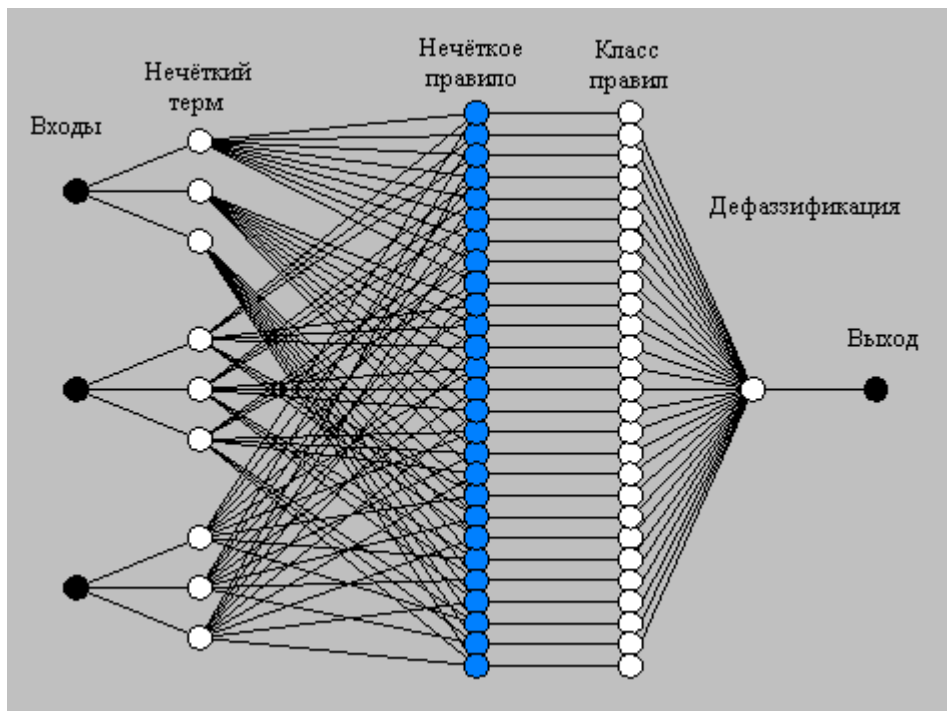


Рисунок 4 – Структура гибридной сети

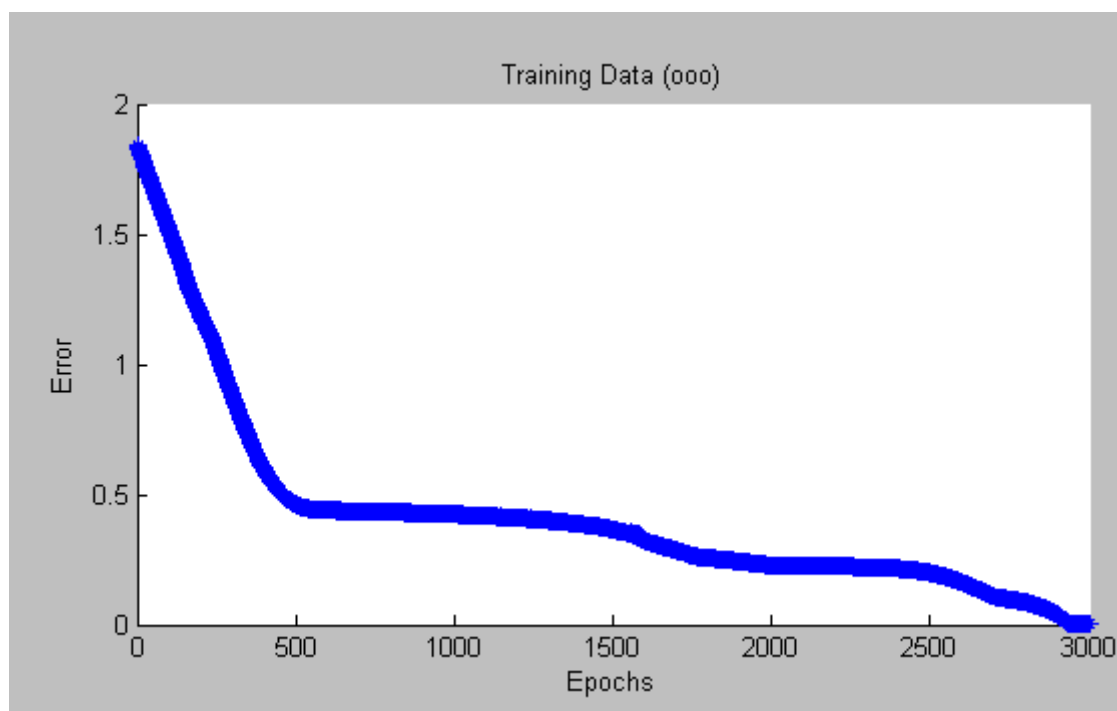


Рисунок 5 – График ошибки обучения

На рисунке 7 представлены диаграммы, соответствующие отклонению процесса по первому входу. Здесь значение температуры в реакторе превышает заданное и равно 155°C (вход 1), остальные параметры остаются в норме; вход 2 – $22,1 \text{ кгс/см}^2$, вход 3 – $0,5 \text{ кгс/см}^2$. Выходное значение равно 1 – это свидетельствует о превышении температуры внутри реактора более, чем 150°C . Системой будут выполнены следующие действия:

1. Прекратится подача бутанола, в реактор будет подан холодный водород.
2. После снижения температуры бутанол будет подаваться в минимальных количествах

Диаграммы, соответствующие отклонению процесса по второму входу представлены на рисунке 8. Выходное значение равное 2 означает, что давление в буферной ёмкости снизилось (менее чем 19 кгс/см^2). В этом случае прекратится подача бутанола на гидрирование, установка будет остановлена и поставлена под давление азота.

На рисунке 9 представлены диаграммы, соответствующие отклонению процесса по третьему входу. Отклонение характеризуется превышением перепада давления «низ-верх» в реакторе более $1,0 \text{ кгс/см}^2$ (вход 3). При этом выходное значение, равное 3, свидетельствует о разрушении катализатора. В этом случае необходимо остановить установку и заменить катализатор.

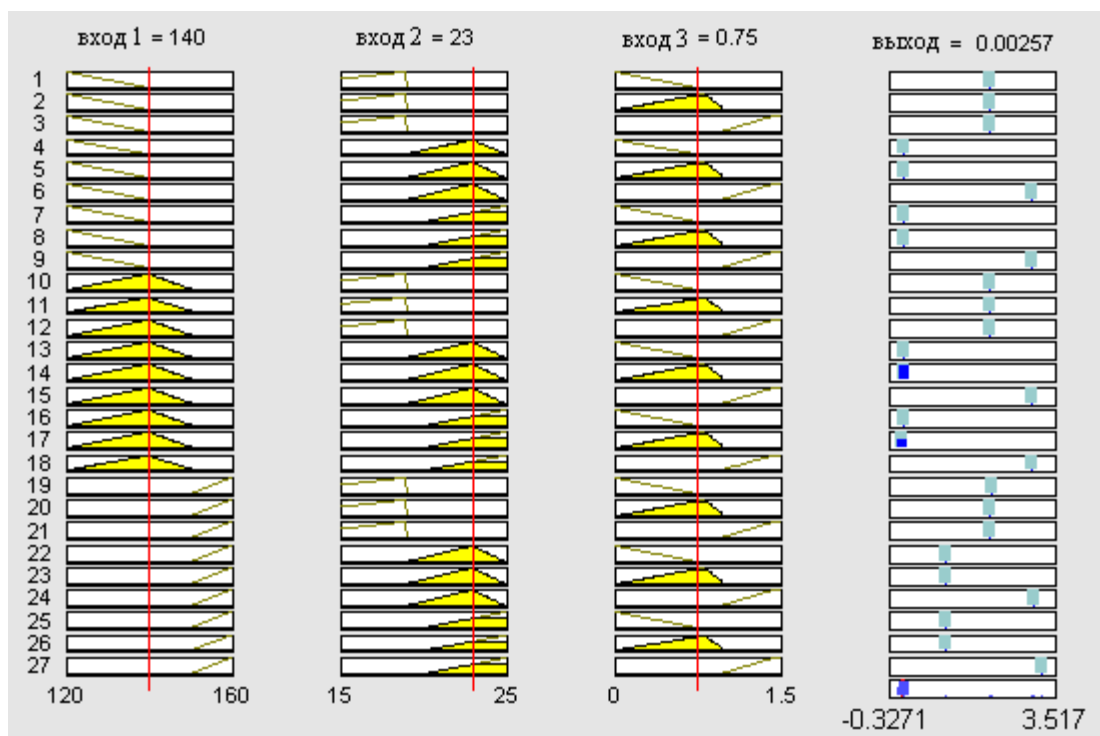


Рисунок 6 – Диаграммы для рабочего режима, соответствующие нормальному течению технологического процесса

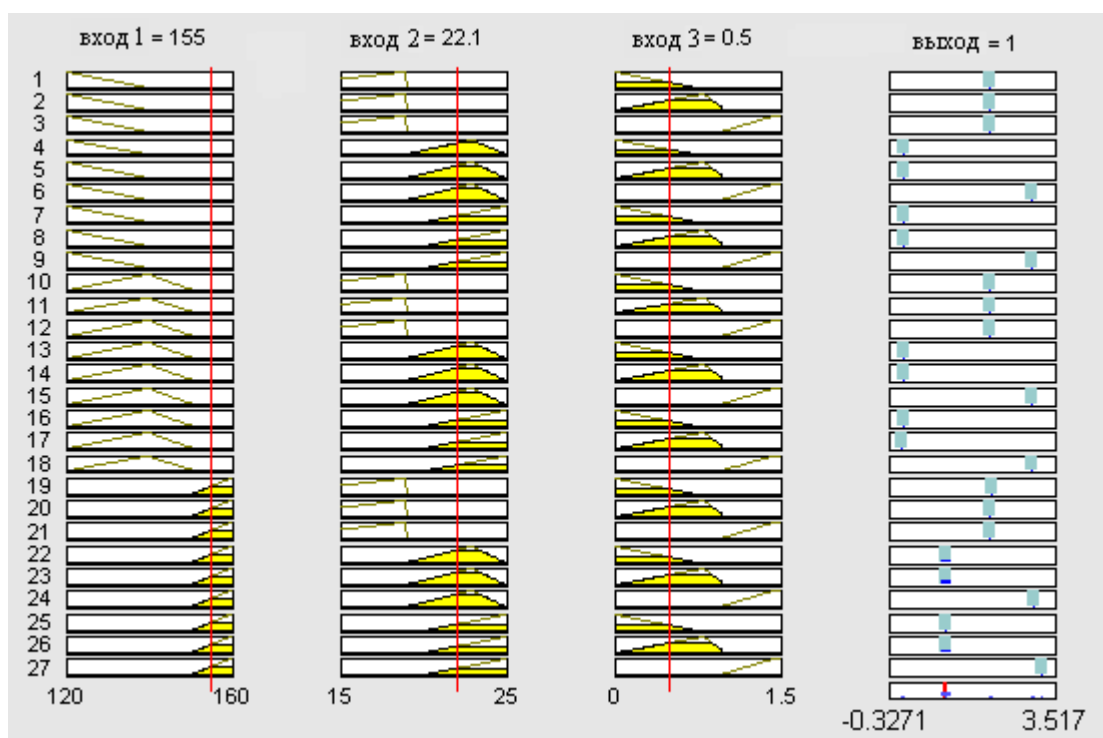


Рисунок 7 – Диаграммы рабочего режима, соответствующие отклонению процесса по первому входу

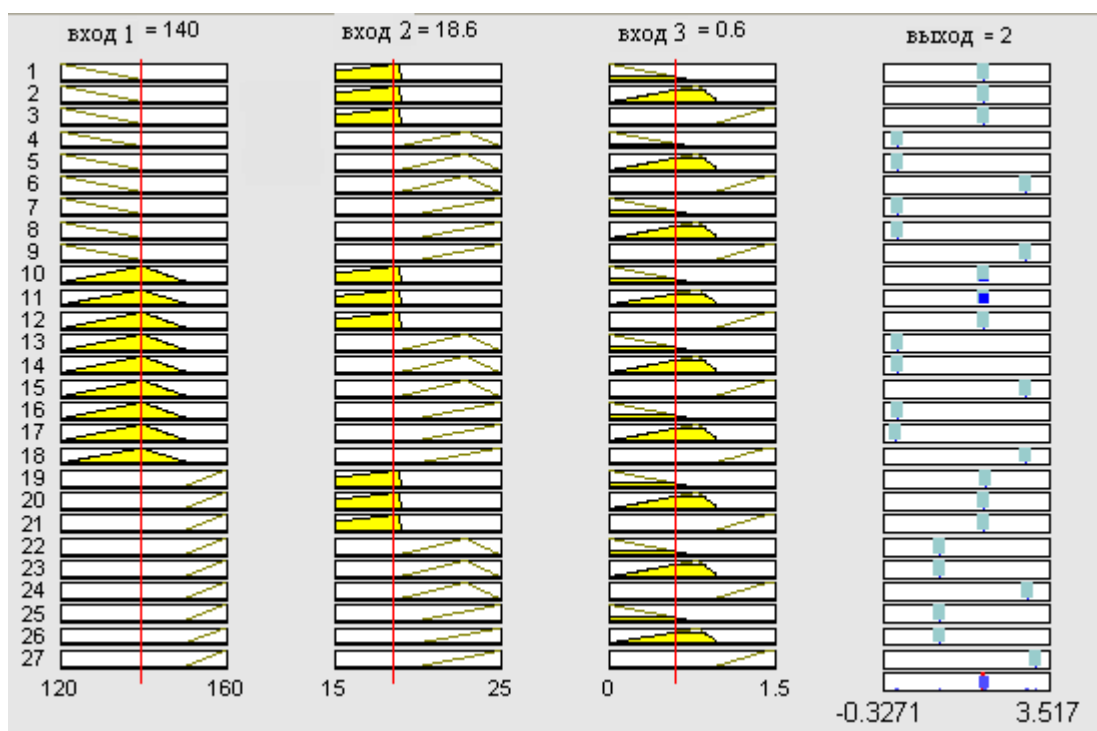


Рисунок 8 – Диаграммы рабочего режима, соответствующие отклонению процесса по второму входу

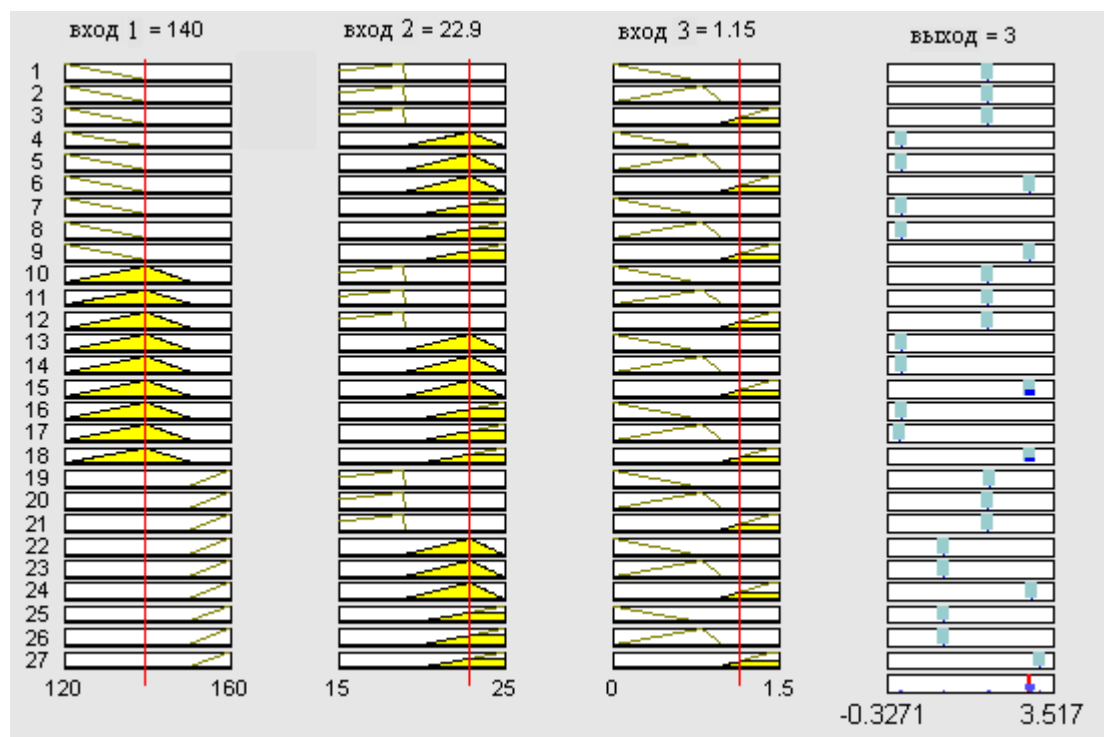


Рисунок 9 – Диаграммы рабочего режима, соответствующие отклонению процесса по третьему входу

Результаты для режима восстановления катализатора представлены в виде диаграмм (рисунки с 10 по 13).

На диаграммах входу «input1» соответствует значение температуры внутри реактора; входу «input2» – температура на входе в реактор; входу «input3» – давление водорода в буферной ёмкости; выход «output» – определяет течение технологического процесса.

Диаграммы рисунка 10 соответствуют течению технологического процесса без отклонений. Значение температуры в реакторе задано 290°C (input1), значение температуры водорода на входе в реактор 290°C (input2), давление водорода в буферной ёмкости – 23 кгс/см² (input3). Выходное значение равно $6,54 \cdot 10^{-211}$ (близко к нулю) – это означает, что технологический процесс проходит без отклонений. Выходное значение близкое к нулю получается и при варьировании входных значений в пределах допускаемых.

На рисунке 11 представлены диаграммы, соответствующие отклонению процесса по первому входу. Значение температуры в реакторе задано 305°C (input1), значение температуры водорода на входе в реактор 285°C (input2), давление водорода в буферной ёмкости – 23 кгс/см² (input3). Выходное значение равно 1 – это свидетельствует о превышении температуры внутри реактора более, чем 300°C. Системой будут поданы управляющие сигналы:

1. На прекращение подачи водорода.
2. На подачу на установку азота.
3. На снижение напряжения на нагрев спирали электроподогревателя.

Диаграммы, соответствующие отклонению процесса по второму входу представлены на рисунке 12. Выходное значение, равное 2, означает, что температура водорода, идущего на реактор, превышает 300°C. В этом случае будет напряжение на нагрев спирали электроподогревателя.

На рисунке 13 представлены диаграммы, соответствующие отклонению процесса по третьему входу. Отклонение характеризуется снижением давления в буферной ёмкости менее 19,0 кгс/см² (input3). При этом выходное значение output=3. В этом случае прекратится подача бутанола на гидрирование, установка будет остановлена и поставлена под давление азота.

Применив пакет расширения FlexTool for MATLAB можно сгенерировать код микроконтроллера для реализации полученной гибридной сети.

Основу программного кода, генерируемого пакетом расширения FlexTool for MATLAB, составляет аппаратно-ориентированное на конкретный тип процессора ядро. Поставляемое с FlexTool for MATLAB программное ядро совместимо с такими контроллерами, как 8096BH, 8096-90, 90196 KR, 90196MC, 80196NT/NQ и др.

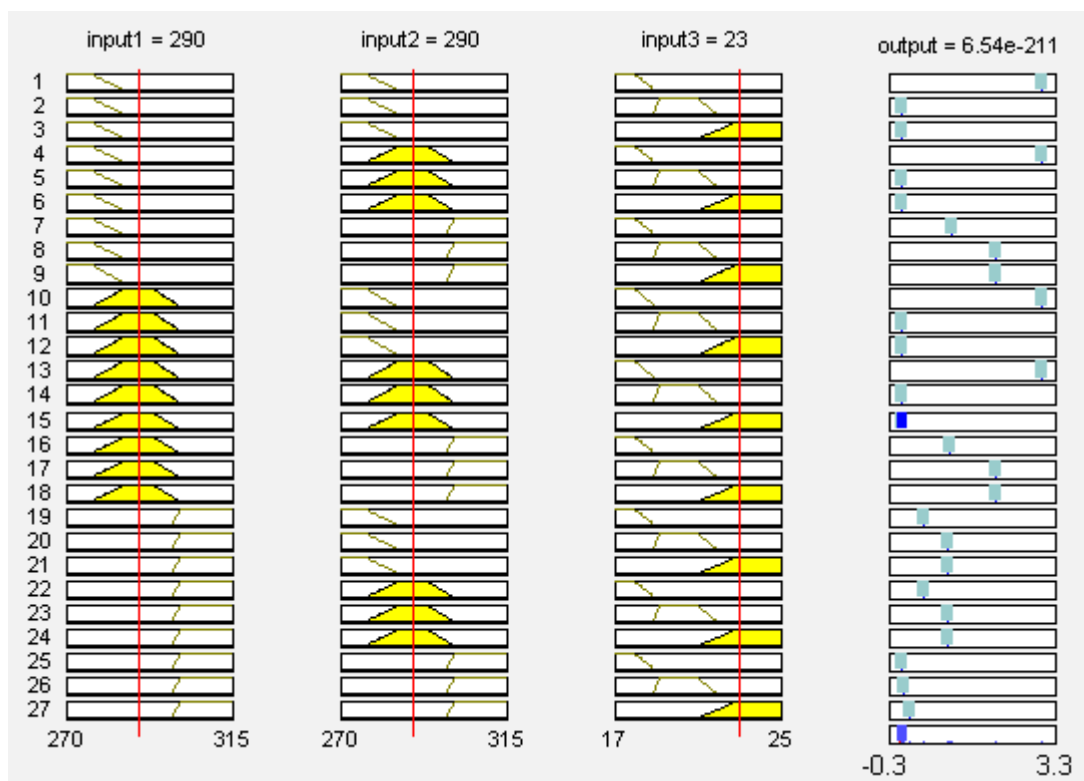


Рисунок 10 – Диаграммы для режима восстановления катализатора при нормальном течении процесса

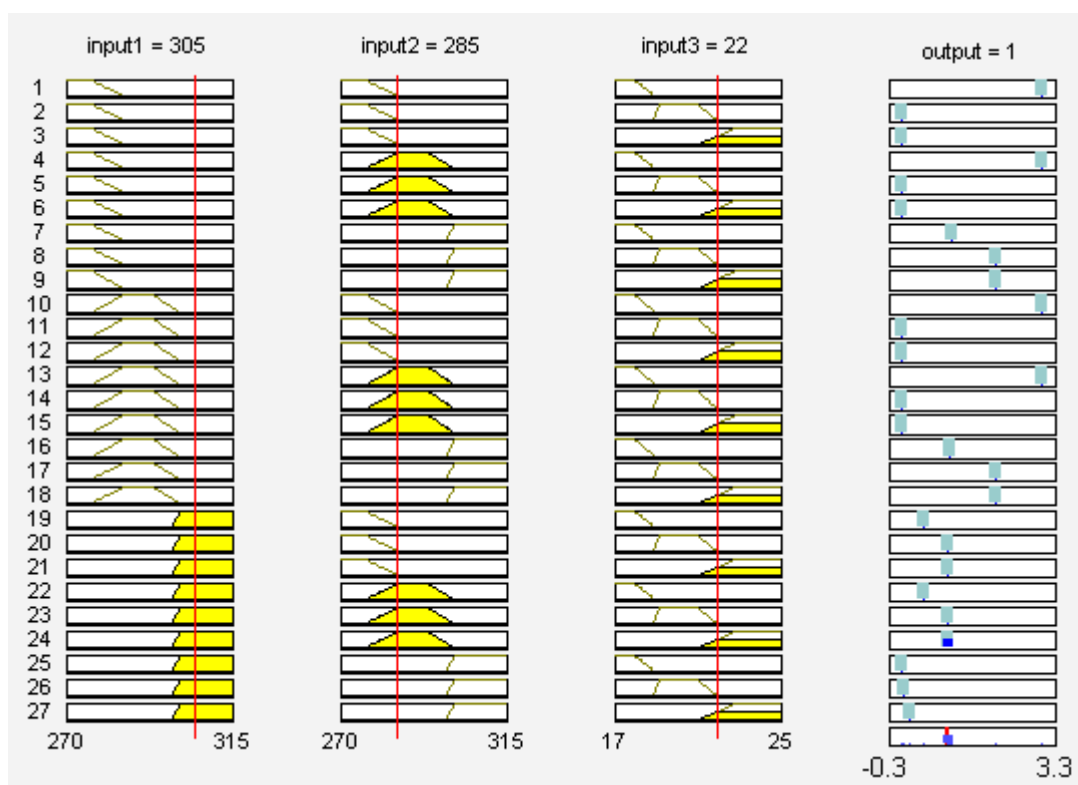


Рисунок 11 – Диаграммы для режима восстановления, соответствующие отклонению процесса по входу «input1»

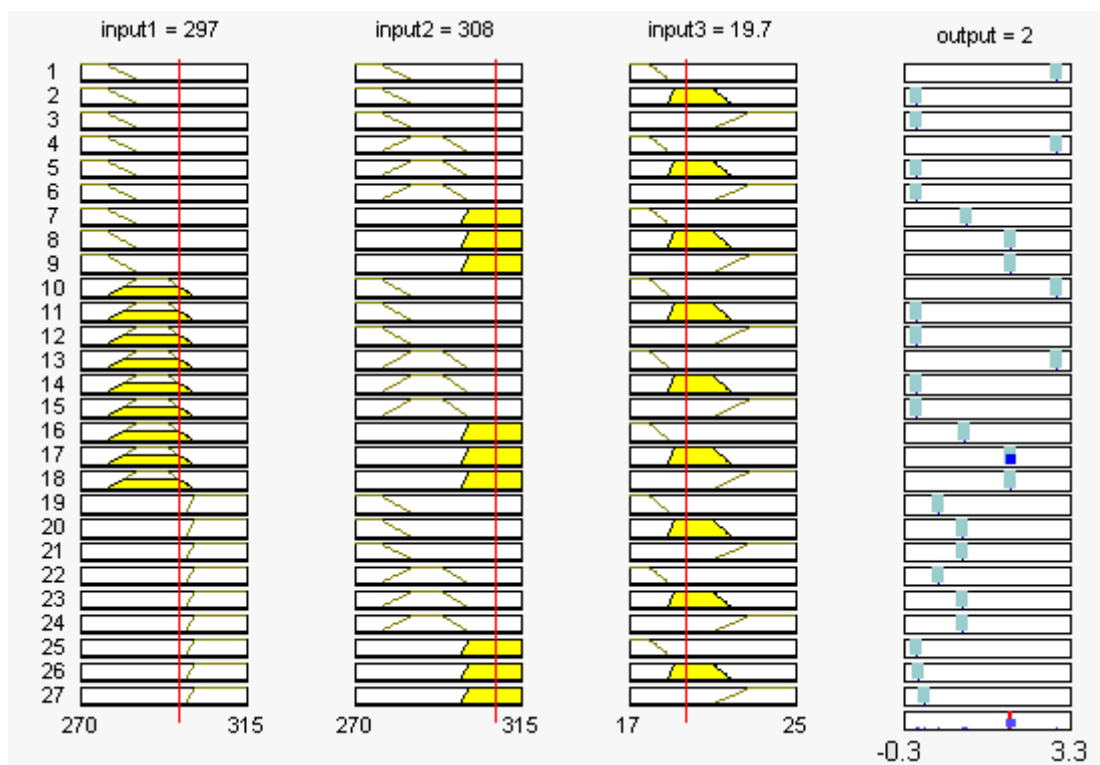


Рисунок 12 – Диаграммы для режима восстановления, соответствующие отклонению процесса по входу «input2»

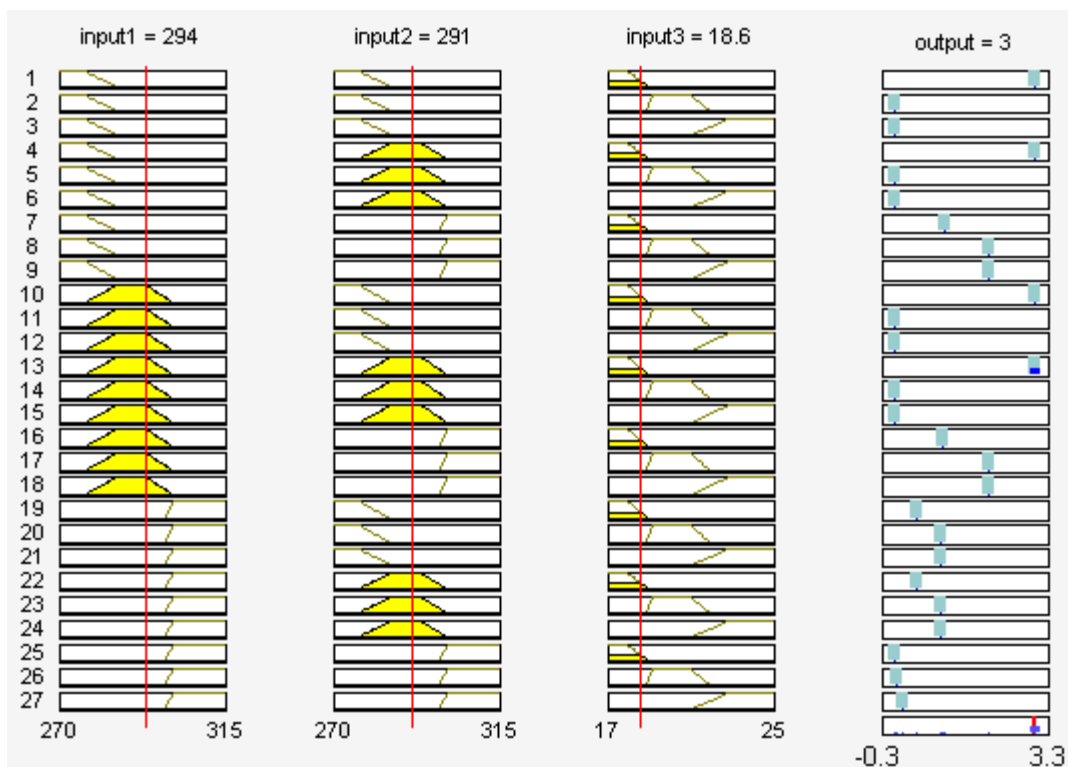


Рисунок 13 – Диаграммы для режима восстановления, соответствующие отклонению процесса по входу «input3»

Тема 3. Экспертные системы

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4

РАЗРАБОТКА НЕЙРО-НЕЧЕТКОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ДИАГНОСТИКЕ РАБОТЫ ПОДШИПНИКОВ КАЧЕНИЯ

Основные теоретические сведения см. лабораторную работу № 3.

Цель – разработать нейро-нечеткий классификатор (команда ANFISedit) режимов работы подшипников качения.

Выбор задания осуществляется по порядковому номеру студента в журнале группы.

Рассмотрим деградацию наружного кольца подшипников качения (рисунок 1).

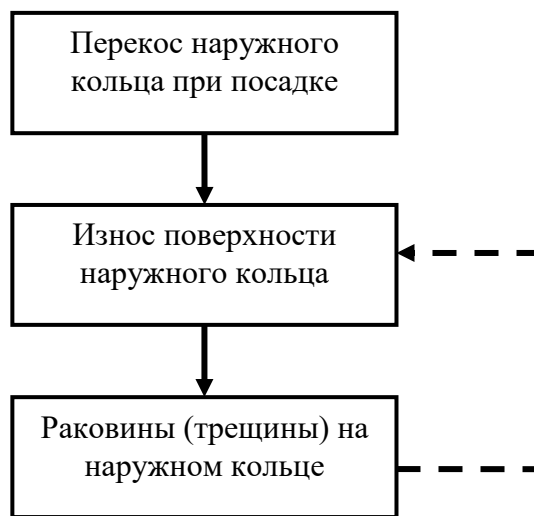


Рисунок 1 – Развитие дефекта подшипника качения

Переко́с наружного кольца подшипника возникает обычно при монтаже подшипника из-за дефектов посадочного места. Он проявляется сразу после монтажа и признаком его является рост составляющих спектра огибающей на частотах kf_n , преимущественно при четных k и, особенно, на второй гармонике $2f_n$. Причиной этого роста является повышенная статистическая нагрузка на поверхности трения в двух противоположных точках наружного кольца. В результате ускоренного износа признаки переко́са могут исчезнуть, однако ускоренный износ наружного кольца продолжится.

Износ наружного кольца подшипника практически всегда происходит локально, изменяя коэффициент трения качения на отдельных участках поверхности наружного кольца. В результате появляется плавная модуляция высокочастотной вибрации частотной f_n и в спектре огибающей вибрации растут гармонические составляющие на частотах kf_n , причем наибольший рост происходит на первой гармонике f_n , а амплитуда кратных гармоник в спектре огибающей быстро падает.

По мере роста износа наружного кольца могут образовываться раковины, а затем они могут сглаживаться и снова представлять собой изношенный участок поверхности плавной формы.

Раковины (трещины) на наружном кольце подшипника приводят к появлению коротких ударных импульсов при контакте каждого тела качения с раковиной (трещиной). В результате появляется ряд гармоник с частотами kf_n в спектре огибающей высокочастотной вибрации, причем число этих гармоник достаточно велико, а их амплитуда слабо снижается с ростом k . Разделить признаки раковины и трещины по результатам анализа спектра огибающей удастся крайне редко и, в основном, за счет более быстрого развития дефекта в случае, если в подшипнике есть трещина. Признаки раковины могут со временем переходить в признаки износа наружного кольца, а признаки трещины стабильны и, более того, растут при каждом последующем измерении.

Таким образом, вид дефекта наружного кольца подшипника качения определяется по частотам кратным частоте перекачивания тел качения по внешней обойме f_n в спектре огибающей. Значения глубин модуляции определяют степень развития дефекта.

Воспользуемся трехуровневой градацией степени дефектности: слабый, средний, сильный. В таблице 1 приведены соответствующие разным уровням развития того или иного дефекта значения глубины амплитудной модуляции.

Таблица 1 – Значения глубины модуляции вибросигнала, %

Вид дефекта	Диагностические уровни дефекта		
	слабый	средний	сильный
1. Перекос	X	Y	Z
2. Износ	X	Y	Z
3. Раковины	X	Y	Z

Кроме перечисленных диагностических признаков в качестве информативного параметра может быть использован рост высокочастотной области спектра. Так при перекосе наружного кольца рост данной области спектра не наблюдается. При износе и появлении раковин (трещин) рост высокочастотной области спектра является дополнительным признаком появления этих дефектов. Использование дополнительного признака повышает достоверность при принятии решения о продолжении эксплуатации оборудования и одновременно позволяет сократить количество используемых признаков (в нашем случае это гармоники $4f_n$, $5f_n$, $6f_n$), что уменьшит затраты на реализацию системы: аппаратные, вычислительные, временные и т.п. Таким образом, определен рабочий словарь признаков

$$U_p = \{m_1, m_2, m_3, РВЧ\},$$

где m_1, m_2, m_3 – значения глубины модуляции соответственно на частотах $f_n, 2f_n, 3f_n$; РВЧ – признак роста высокочастотной области спектра огибающей вибросигнала, принимает два значения: 1 (есть рост) и 0 (отсутствие роста).

Составим алфавит классов $\mathbf{W}$ и описание каждого класса, т.е. зависимости вида

$$W_i = p_i(m_1, m_2, m_3, \text{РВЧ}).$$

Тогда возможны следующие варианты, отображенные на рисунке 2.

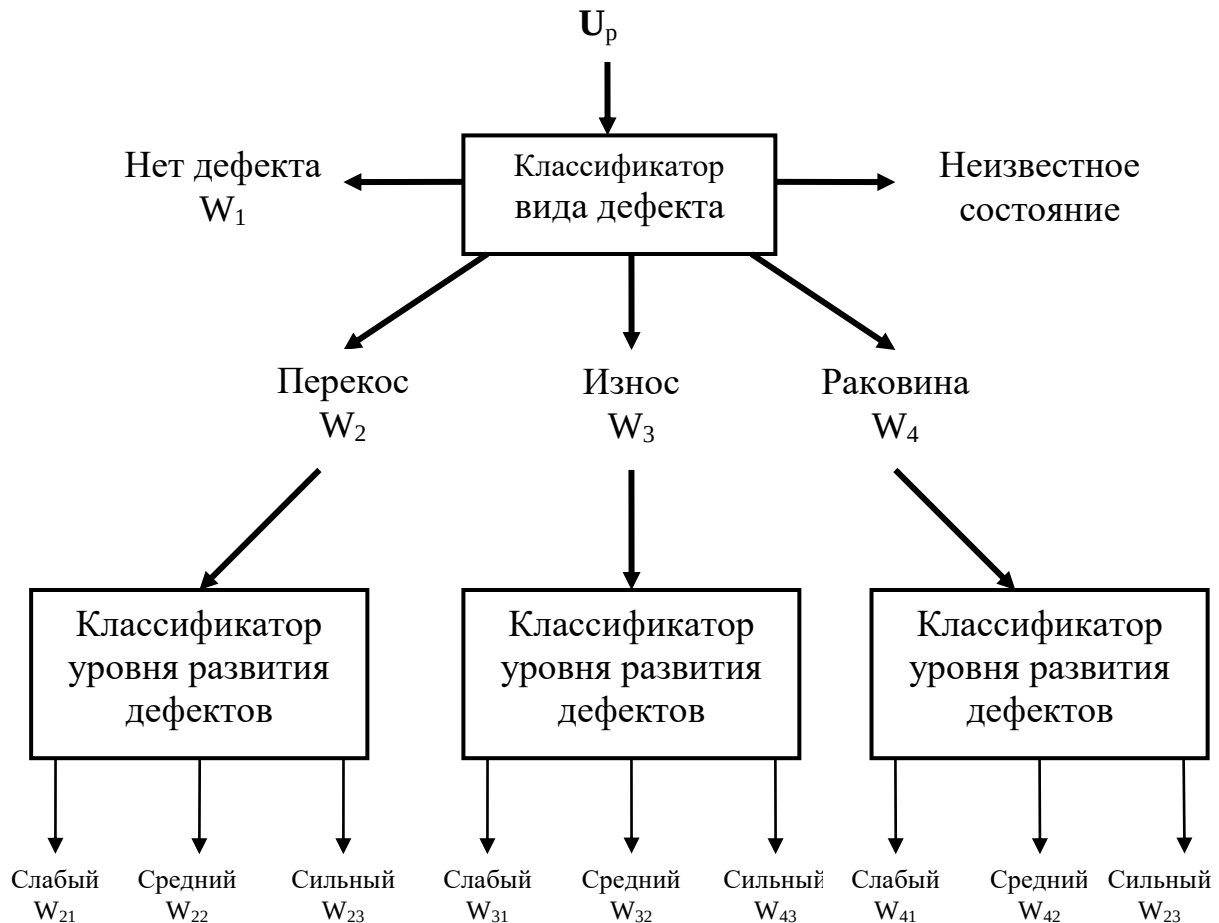


Рисунок 2 – Исходы процесса диагностической классификации

В соответствии с рисунком 2 и таблицей 1 составим базу знаний диагностической системы:

W_1 : если $m_1 < X$ И $m_2 < X$ И $m_3 < X$ И РВЧ = 0;

W_{21} : если $m_3 < m_1 < m_2$ И $X \leq m_2 < Y$ И РВЧ = 0;

W_{22} : если $m_3 < m_1 < m_2$ И $Y \leq m_2 < Z$ И РВЧ = 0;

W_{23} : если $m_3 < m_1 < m_2$ И $Z \leq m_2$ И РВЧ = 0;

W_{31} : если $m_3 < m_2 < m_1$ И $X \leq m_1 < Y$ И РВЧ = 1;

W_{32} : если $m_3 < m_2 < m_1$ И $Y \leq m_1 < Z$ И РВЧ = 1;

W_{33} : если $m_3 < m_2 < m_1$ И $Z \leq m_1$ И РВЧ = 1;

W_{41} : если $m_3 \approx m_2 \approx m_1$ И $X \leq m_1 < Y$ И РВЧ = 1;

W_{42} : если $m_3 \approx m_2 \approx m_1$ И $Y \leq m_1 < Z$ И РВЧ = 1;

W₄₃: если $m_3 \approx m_2 \approx m_1$ И $Z \leq m_1$ И РВЧ = 1;

W₅: другие комбинации $m_i \geq 4$ совместно с РВЧ.

Из приведенной базы знаний видно, что тип дефекта определяется из соотношения между значениями глубин модуляций кратных f_n частот. Дополнительную информацию несет параметр РВЧ. Степень развития дефекта определяется значениями модуляции на частотах f_n и $2 f_n$. Наличие класса W₅ позволяет накапливать знания в процессе эксплуатации системы диагностики. Следовательно, задача классификации может быть разбита на два параллельных процесса: определение типа и величины дефекта. В первом случае используются все четыре признака, а во втором – только два (m_1, m_2). Задача отнесения дефекта к одному из возможных типов имеет 4 решения (W₂, W₃, W₄, W₅). Задача определения степени развития дефекта – 3 решения (W_{i1}, W_{i2}, W_{i3}). Определение факта наличия или отсутствия дефекта наружного кольца (т.е. отнесение к классу W₁ или нет) предшествует указанным задачам и выступает определителем необходимости дальнейшего исследования.

Таблица 2 – Значения X, Y, Z для выполнения контрольной работы

№ варианта	X	Y	Z	№ варианта	X	Y	Z
1	2	4	6	7	7	10	13
2	4	8	16	8	9	15	20
3	5	10	15	9	5	15	30
4	3	9	18	10	8	14	19
5	6	10	15	11	1	5	9
6	10	20	30	12	7	11	15

Тема 3. Экспертные системы

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

РАЗРАБОТКА НЕЙРО-НЕЧЕТКОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ДИАГНОСТИКЕ РЕЖИМОВ РАБОТЫ ТУРБОГЕНЕРАТОРА

Цель – изучение методов разработки нейро-нечеткой системы поддержки принятия решений при диагностике работы турбогенератора

Вибрационные параметры турбогенератора могут служить для диагностики его состояния. Фактически по параметрам колебаний подшипников можно судить о следующих нарушениях (таблица 1).

Таблица 1 – Причины и признаки распространенных нарушений в работе турбогенератора

Основные причины		Скорость нарастания	Частота
Динамическая не-уравновешенность ротора	Обрыв лопаток и бандажей, разрушение дисков, задевания в проточной части	Быстрая	Оборотная
	Неравномерный износ лопаток, неравномерный занос солями лопаточного аппарата	Медленная	Оборотная
	Тепловая разбалансировка, термическая нестабильность и неоднородность металла, неправильные режимы пуска и останова	Средняя	Оборотная
Расцентровка ротора	Тепловая нестабильность	Средняя	Оборотная
	Появление трещин в фундаменте	Средняя	Оборотная+ высокочастотные гармоники
Потеря устойчивости вала на масляной пленке	Уменьшение удельного давления на подшипник в процессе эксплуатации	Медленная	Половинная
	Заниженная температура масла на входе в подшипник	Средняя	Половинная
Появление возмущающих сил электромагнитного происхождения	Витковые замыкания в роторе	Быстрая	Оборотная+ двойная оборотная

Примечание: оборотная частота – 50 Гц, скорость нарастания: быстрая – порядка 1-2 мм/с в минуту, средняя – 1-2 мм/с за срок до 7 суток, медленная – от 7 до месяца и выше.

Правилами технической эксплуатации электрических станций и сетей регламентируются вибрационные параметры турбины. Причем с каждой редакцией правил эти требования ужесточаются.

Современная классификация вибрации турбогенераторов основана на понятии средней квадратической виброскорости (или эквивалентной). Средняя квадратическая виброскорость, возве-

денная в квадрат, есть сумма квадратов произведений угловых частот гармоник колебаний на их амплитуды. Преобладающими составляющими вибрации турбогенераторов являются оборотная, половинная и двойная оборотная частота. Для половинной частоты максимально допустимой является виброскорость порядка 1 мм/с, для оборотной – 9 мм/с (при наличии других гармоник), для двойной оборотной – 6,7 мм/с. Для моделирования нейросетевого алгоритма нормируем эти параметры к оптимальной виброскорости (максимальная виброскорость для каждой частоты приблизительно в 2,5 раза больше оптимальной). Составим таблицу классификатора, которая содержит девять кластеров (таблица 2).

Таблица 2 – База знаний диагностической системы

Виброскорость Частота		Оптимально	Допустимо	Недопустимо
~25 Гц	абсолют.	0,4 мм/с	1 мм/с	>1 мм/с
	нормир.	1	2,5	>2,5
~50 Гц	абсолют.	3,6 мм/с	9 мм/с	>9 мм/с
	нормир.	1	2,5	>2,5
~100 Гц	абсолют.	2,68 мм/с	6,7 мм/с	>6,7 мм/с
	нормир.	1	2,5	>2,5

Вибрация должна замеряться в трех направлениях: вертикальном, горизонтально-продольном и горизонтально-поперечном. Если вибрация хотя бы одного из подшипников в одном из трех направлений превышает значение «допустимо» для данного типа машин, то вибрационное состояние всего агрегата признается неудовлетворительным, и турбогенератор должен быть выведен в ремонт для устранения вибрации.

Зададим обучающую выборку. Для этого создадим модуль, в котором при помощи функции **nnngenc**, позволяющей создавать необходимое количество кластеров с заданным разбросом параметров и случайным распределением точек, сгенерируем девять кластеров, соответствующих характерным частотам вибраций турбогенераторов и значениям виброскоростей (таблица 2). Получим следующую последовательность:

P=[0,5311 0,5832 0,5022 0,3707 0,5958 0,3877 0,6246 0,5282 0,4957 0,4155
0,5311 0,5832 0,5022 0,3707 0,5958 0,3877 0,6246 0,5282 0,4957 0,4155 0,5311
0,5832 0,5022 0,3707 0,5958 0,3877 0,6246 0,5282 0,4957 0,4155 0,9719 0,9730
0,9169 1,0793 0,9177 0,9533 0,9213 0,8362 1,1332 1,0328 0,9719 0,9730 0,9169

1,0793	0,9177	0,9533	0,9213	0,8362	1,1332	1,0328	0,9719	0,9730	0,9169	1,0793
0,9177	0,9533	0,9213	0,8362	1,1332	1,0328	2,1056	1,9203	1,9170	1,9307	1,8772
1,7572	2,0188	2,1418	2,1037	1,8405	2,1056	1,9203	1,9170	1,9307	1,8772	1,7572
2,0188	2,1418	2,1037	1,8405	2,1056	1,9203	1,9170	1,9307	1,8772	1,7572	2,0188
2,1418	2,1037	1,8405;	0,1483	0,6202	0,6428	0,3491	0,5285	0,4410	0,7640	0,8560
0,1939	0,2013	1,7478	1,7350	1,8647	1,5089	1,9427	2,2795	1,7662	1,5583	1,7375
1,5800	2,9427	2,8751	3,2671	2,6334	3,0735	2,9523	3,0929	2,8858	3,0301	2,9210
0,1483	0,6202	0,6428	0,3491	0,5285	0,4410	0,7640	0,8560	0,1939	0,2013	1,7478
1,7350	1,8647	1,5089	1,9427	2,2795	1,7662	1,5583	1,7375	1,5800	2,9427	2,8751
3,2671	2,6334	3,0735	2,9523	3,0929	2,8858	3,0301	2,9210	0,1483	0,6202	0,6428
0,3491	0,5285	0,4410	0,7640	0,8560	0,1939	0,2013	1,7478	1,7350	1,8647	1,5089
1,9427	2,2795	1,7662	1,5583	1,7375	1,5800	2,9427	2,8751	3,2671	2,6334	3,0735
2,9523	3,0929	2,8858	3,0301	2,9210						

Графическое представление полученной последовательности представлено на рисунке 1.

Данную последовательность сохраним в файл **Data.mat**. Как видно, обучающая выборка представлена матрицей, которая содержит два столбца. А командой **load Data** считаем данные обучающей выборки из файла **Data.mat**.

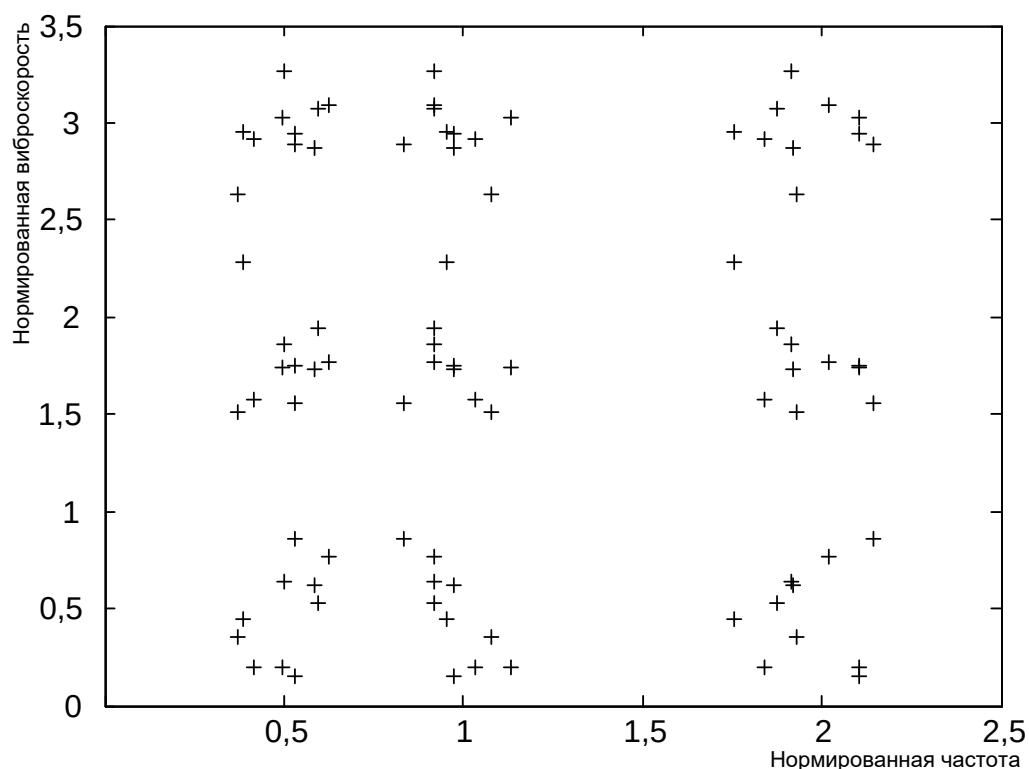


Рисунок 1 – Исходные данные

Задание на контрольную работу в соответствии с таблицей 3 представлено в таблице 4.

Таблица 3 – База знаний

Виброскорость Частота		Оптимально	Допустимо	Недопустимо
~25 Гц	абсолют.	X1	Y1	> Y1
~50 Гц	абсолют.	X2	Y2	> Y2
~100 Гц	абсолют.	X3	Y3	> Y3

Таблица 4 – Варианты заданий

№ варианта	X1	X2	X3	Y1	Y2	Y3	№ варианта	X1	X2	X3	Y1	Y2	Y3
13	1	3	5	3	6	9	21	0,5	1	2,5	1,2	3	6
14	2	4	6	4	7	10	22	1,7	2,2	3,4	4	6	8
15	3	5	7	6	10	14	23	0,8	5	9	2,5	10	16
16	4	8	15	8	16	28	24	2,1	4	5,5	5	8	11
17	5	10	14	10	20	29	25	1,5	3,5	4,2	3,2	8	10
18	6	10	18	12	19	32	26	3,4	4,8	5,5	8	9	12
19	7	10	17	14	19	30	27	0,3	1,1	2,6	2	3	6
20	8	15	20	16	25	34	28	3	6	9	6	12	18

СПИСОК ЛИТЕРАТУРЫ

1. В.Дьяконов, И.Абраменкова, В.Круглов. MATLAB с пакетами расширений под ред. Проф. В.П. Дьяконова. – М.: Нолидж. – 2001. – 880с.
2. Дьяконов В. MATHCAD 8/2000: Специальный справочник – СПб.: Питер, 2001. – 592 с.
3. Дьяконов В., Круглов В. Математические пакеты расширения MATLAB: Специальный справочник. – СПб.: Питер, 2001. – 480 с.
4. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ – Питербург, 2003. – 736 с.
5. Макаров Е.Г. Инженерные расчеты в MathCAD: Учебный курс. – СПб.: Питер, 2003. – 448 с.
6. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. - М.: Наука, 1986. -312 с.
7. Потемкин В.Г.Вычисления в среде MATLAB.– М.: Диалог - МИФИ, 2004. – 720 с.
8. Трахтенгерц Э.А. Компьютерная поддержка принятия решений. - М.: СИНТЕГ, 1998. – 376 с.
9. www.mathworks.com [Электронный ресурс]: Фирма The Mathworks, Inc. (USA) – Режим доступа: [http:// www.mathworks.com](http://www.mathworks.com), свободный. – Загл. с экрана. – Яз. англ.
10. www.matlab.ru [Электронный ресурс]: Консультационный Центр MATLAB – Режим доступа: <http://www.matlab.ru>, свободный. – Загл. с экрана. – Яз. русский, англ.
11. www.mathsoft.com [Электронный ресурс]: Фирма Mathsoft Engineering & Education, Inc. (USA) – Режим доступа: <http://support.mathsoft.com>, свободный. – Загл. с экрана. – Яз. англ.
12. www.basegroup.ru. [Электронный ресурс]: Фирма BASEGROUP Labs (Россия) – Режим доступа: <http://www.basegroup.ru>, свободный. – Загл. с экрана. – Яз. русский, англ.
13. Галушкин А. И. Нейрокомпьютеры. Учеб. пособие для вузов/ Общая ред. А. И. Галушкина. – М.: ИПРЖР, 2000. – 528 с.
14. Головкин В. А. Нейронные сети: обучение, организация и применение. Кн. 4: Учеб. пособие для вузов/ Общая ред. А. И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.
15. Калан Р. Основные концепции нейронных сетей: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 288 с.
16. Комарцова Л. Г., Максимов А. В. Нейрокомпьютеры: Учеб. Пособие для вузов. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2002. – 320 с.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение
высшего образования**
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
НЕВИННОМЫССКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ
(ФИЛИАЛ)»**

Методические указания к самостоятельной работе
для студентов направления
15.03.04 «Автоматизация технологических процессов и производств»
по дисциплине
«ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ»

Невинномысск, 2026

Методические указания разработаны в соответствии с требованиями ФГОС ВО в части содержания и уровня подготовки выпускников направления подготовки 15.03.04 «Автоматизация технологических процессов и производств».

Методические указания содержат рекомендации по организации самостоятельной работы студента при изучении дисциплины «Интеллектуализация систем управления».

Составитель

доцент кафедры ИСЭА А.А. Евдокимов

Ответственный редактор

доцент кафедры ИСЭА Д.В. Болдырев

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ»	5
1.1. Подготовка к лекциям.....	7
1.2. Подготовка к лабораторным занятиям.....	8
1.3. Подготовка к практическим занятиям	9
1.4. Самостоятельное изучение материала тем	11
2. СРЕДСТВА ОЦЕНИВАНИЯ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ»	15
3. ОТЧЕТНОСТЬ ПО ДИСЦИПЛИНЕ	20
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	21

ВВЕДЕНИЕ

Дисциплина «Интеллектуализация систем управления» ставит своей целью формирование следующих компетенций будущего бакалавра по направлению подготовки 15.03.04 — Автоматизация технологических процессов и производств.

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.	ИД-3 _{УК-2} Обеспечивает выполнение проекта в соответствии с установленными целями, сроками и затратами, исходя из действующих правовых норм, имеющихся ресурсов и ограничений, в том числе с использованием цифровых инструментов.	Определяет параметры средств и систем автоматизации, обеспечивающие оптимальность проектных решений
ПК-2. Способен участвовать в работах по расчету и проектированию средств и систем автоматизации с использованием современных информационных технологий, методов и средств проектирования.	ИД-1 _{ПК-2} Рассчитывает и проектирует средства и системы автоматизации в соответствии с техническим заданием.	Рассчитывает и проектирует средства и системы автоматизации с использованием технологий искусственного интеллекта.
	ИД-3 _{ПК-2} Выполняет сбор и анализ исходных данных для расчета и проектирования средств и систем управления с использованием современных информационных технологий.	Выполняет сбор и анализ исходных данных для расчета и проектирования средств и систем управления с использованием технологий искусственного интеллекта.

Главными задачами дисциплины являются: усвоение студентами основных положений теории искусственного интеллекта, способов представления знаний и их инженерии, правил анализа и синтеза экспертных систем.

В результате освоения дисциплины студент должен:

- знать основные положения теории искусственного интеллекта; способы организации интеллектуальных систем; способы построения баз данных, баз знаний и экспертных систем; модели и методы формализации и представления знаний в интеллектуальных системах;
- уметь использовать интеллектуальные информационные системы, инструментальные средства управления базами данных и знаний;
- владеть навыками решения задач с помощью систем искусственного интеллекта и экспертных систем.

Методические указания предназначены для выполнения самостоятельной работы по дисциплине «Интеллектуализация систем управления» с учетом требований ФГОС ВО для направления подготовки 15.03.04 — Автоматизация технологических процессов и производств. Они способствуют лучшему усвоению студентами теоретических положений и обеспечивает приобретение практических навыков по исследованию элементов и систем автоматического регулирования и управления.

1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ»

Самостоятельная работа студентов (далее — СРС) является неотъемлемой составляющей образовательного процесса в Университете и является обязательной для каждого студента. Основная цель СРС — освоение в полном объеме образовательной программы и последовательное формирование компетенций эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности. Самостоятельная работа кон-

кретна по своей предметной направленности и сопровождается непрерывным контролем и оценкой ее результатов.

Количество часов, отводимое на самостоятельную работу, определяется учебным планом направления подготовки 15.03.04.

Содержательно самостоятельная работа студентов определяется ФГОС ВО направления подготовки 15.03.04, программой и учебно-методическим комплексом дисциплины «Интеллектуализация систем управления».

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Интеллектуализация систем управления», объема часов на ее изучение, вида заданий для СРС, индивидуальных возможностей студентов и условий учебной деятельности.

Формы самостоятельной работы студентов определяются содержанием дисциплины «Интеллектуализация систем управления», степенью подготовленности студентов. Они могут быть тесно связаны с теоретическим курсом и иметь учебный или учебно-исследовательский характер. Форму самостоятельной работы студентов определяют кафедра ИСЭА при разработке программы дисциплины «Интеллектуализация систем управления».

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

СРС, не предусмотренная образовательной программой, учебным планом и учебно-методическими материалами, раскрывающими и конкретизирующими их содержание, осуществляется студентами инициативно, с целью реализации собственных учебных и научных интересов.

В учебном процессе выделяют аудиторную и внеаудиторную самостоятельную работу.

Аудиторная самостоятельная работа по дисциплине «Интеллектуализация систем управления» выполняется на учебных занятиях (лекциях, практических, лабораторных занятиях и консультациях) под руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа студентов выполняется во внеаудиторное время по заданию и при методическом руководстве и контроле преподавателя, но без его непосредственного участия. СРС включает в себя:

- подготовку к аудиторным занятиям (лекционным и практическим) и выполнение соответствующих заданий;
- работу над отдельными темами учебных дисциплин (модулей) в соответствии с учебно-тематическими планами;
- выполнение контрольных работ;
- подготовку ко всем видам промежуточных и итоговых контрольных испытаний.

1.1. Подготовка к лекциям

Главное в период подготовки к лекционным занятиям — научиться методам самостоятельного умственного труда, сознательно развивать свои творческие способности и овладевать навыками творческой работы. Для этого необходимо строго соблюдать дисциплину учебы и поведения. Четкое планирование своего рабочего времени и отдыха является необходимым условием для успешной самостоятельной работы. В основу его нужно положить рабочие программы изучаемых в семестре дисциплин.

Каждому студенту следует составлять еженедельный и семестровый планы работы, а также план на каждый рабочий день. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подводить итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине это произошло. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана.

Слушание и запись лекций — сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях.

Конспект лекций лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, определения, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек. Лучше если они будут собственными, чтобы не приходилось просить их у однокурсников и тем самым не отвлекать их во время лекции. Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть знаниями.

1.2. Подготовка к лабораторным занятиям

Для того чтобы лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по рассмотренному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что

только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться студентом на лабораторных занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

1.3. Подготовка к практическим занятиям

Подготовку к каждому практическому занятию студент должен начать с ознакомления с методическими указаниями, которые включают содержание работы. Тщательное продумывание и изучение вопросов основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. На основе ин-

дивидуальных предпочтений студенту необходимо самостоятельно выбрать тему доклада по проблеме и по возможности подготовить по нему презентацию.

Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции (устно или письменно). Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы семинара, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

В зависимости от содержания и количества отведенного времени на изучение каждой темы практическое занятие может состоять из четырех-пяти частей:

1. Обсуждение теоретических вопросов, определенных программой дисциплины.
2. Доклад и/или выступление с презентациями по выбранной проблеме.
3. Обсуждение выступлений по теме — дискуссия.
4. Выполнение практического задания с последующим разбором полученных результатов или обсуждение практического задания.
5. Подведение итогов занятия.

Первая часть — обсуждение теоретических вопросов — проводится в виде фронтальной беседы со всей группой и включает выборочную проверку преподавателем теоретических знаний студентов. Примерная продолжительность — до 15 минут. Вторая часть — выступление студентов с докладами, которые должны сопровождаться презентациями с целью усиления наглядности восприятия, по одному из вопросов практического занятия. Обязательный элемент доклада — представление и анализ статистических данных, обоснование социальных последствий любого экономического факта, явле-

ния или процесса. Примерная продолжительность — 20-25 минут. После докладов следует их обсуждение — дискуссия. В ходе этого этапа практического занятия могут быть заданы уточняющие вопросы к докладчикам. Примерная продолжительность — до 15-20 минут. Если программой предусмотрено выполнение практического задания в рамках конкретной темы, то преподавателями определяется его содержание и дается время на его выполнение, а затем идет обсуждение результатов. Подведением итогов заканчивается практическое занятие.

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

1.4. Самостоятельное изучение материала тем

Конспект — наиболее совершенная и наиболее сложная форма записи. Слово «конспект» происходит от латинского «conspectus», что означает «обзор, изложение». В правильно составленном конспекте обычно выделено самое основное в изучаемом тексте, сосредоточено внимание на наиболее су-

шественном, в кратких и четких формулировках обобщены важные теоретические положения.

Конспект представляет собой относительно подробное, последовательное изложение содержания прочитанного. На первых порах целесообразно в записях ближе держаться тексту, прибегая зачастую к прямому цитированию автора. В дальнейшем, по мере выработки навыков конспектирования, записи будут носить более свободный и сжатый характер.

Конспект книги обычно ведется в тетради. В самом начале конспекта указывается фамилия автора, полное название произведения, издательство, год и место издания. При цитировании обязательная ссылка на страницу книги. Если цитата взята из собрания сочинений, то необходимо указать соответствующий том. Следует помнить, что четкая ссылка на источник — непереносимое правило конспектирования. Если конспектируется статья, то указывается, где и когда она была напечатана.

Конспект подразделяется на части в соответствии с заранее продуманным планом. Пункты плана записываются в тексте или на полях конспекта. Писать его рекомендуется четко и разборчиво, так как небрежная запись с течением времени становится малопонятной для ее автора. Существует правило: конспект, составленный для себя, должен быть по возможности написан так, чтобы его легко прочитал и кто-либо другой.

Формы конспекта могут быть разными и зависят от его целевого назначения (изучение материала в целом или под определенным углом зрения, подготовка к докладу, выступлению на занятии и т.д.), а также от характера произведения (монография, статья, документ и т.п.). Если речь идет просто об изложении содержания работы, текст конспекта может быть сплошным, с выделением особо важных положений подчеркиванием или различными значками.

В случае, когда не ограничиваются переложением содержания, а фиксируют в конспекте и свои собственные суждения по данному вопросу или дополняют конспект соответствующими материалами их других источников,

следует отводить место для такого рода записей. Рекомендуется разделить страницы тетради пополам по вертикали и в левой части вести конспект произведения, а в правой свои дополнительные записи, совмещая их по содержанию.

Конспектирование в большей мере, чем другие виды записей, помогает вырабатывать навыки правильного изложения в письменной форме важные теоретических и практических вопросов, умение четко их формулировать и ясно излагать своими словами.

Таким образом, составление конспекта требует вдумчивой работы, затраты времени и труда. Зато во время конспектирования приобретаются знания, создается фонд записей.

Конспект может быть текстуальным или тематическим. В текстуальном конспекте сохраняется логика и структура изучаемого произведения, а запись ведется в соответствии с расположением материала в книге. За основу тематического конспекта берется не план произведения, а содержание какой-либо темы или проблемы.

Текстуальный конспект желательно начинать после того, как вся книга прочитана и продумана, но это, к сожалению, не всегда возможно. В первую очередь необходимо составить план произведения письменно или мысленно, поскольку в соответствии с этим планом строится дальнейшая работа. Конспект включает в себя тезисы, которые составляют его основу. Но, в отличие от тезисов, конспект содержит краткую запись не только выводов, но и доказательств, вплоть до фактического материала. Иначе говоря, конспект — это расширенные тезисы, дополненные рассуждениями и доказательствами, мыслями и соображениями составителя записи.

Как правило, конспект включает в себя и выписки, но в него могут войти отдельные места, цитируемые дословно, а также факты, примеры, цифры, таблицы и схемы, взятые из книги. Следует помнить, что работа над конспектом только тогда будет творческой, когда она не ограничена текстом

изучаемого произведения. Нужно дополнять конспект данными из других источников.

В конспекте необходимо выделять отдельные места текста в зависимости от их значимости. Можно пользоваться различными способами: подчеркиваниями, вопросительными и восклицательными знаками, репликами, краткими оценками, писать на полях своих конспектов слова: «важно», «очень важно», «верно», «характерно».

В конспект могут помещаться диаграммы, схемы, таблицы, которые придадут ему наглядность.

Составлению тематического конспекта предшествует тщательное изучение всей литературы, подобранной для раскрытия данной темы. Бывает, что какая-либо тема рассматривается в нескольких главах или в разных местах книги. А в конспекте весь материал, относящийся к теме, будет сосредоточен в одном месте. В плане конспекта рекомендуется делать пометки, к каким источникам (вплоть до страницы) придется обратиться для раскрытия вопросов. Тематический конспект составляется обычно для того, чтобы глубже изучить определенный вопрос, подготовиться к докладу, лекции или выступлению на семинарском занятии. Такой конспект по содержанию приближается к реферату, докладу по избранной теме, особенно если включает и собственный вклад в изучение проблемы.

2. СРЕДСТВА ОЦЕНИВАНИЯ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ИНТЕЛЛЕКТУАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ»

Вопросы для собеседования

5 семестр

1. Дайте определение искусственному интеллекту.
2. Какие три направления сложились при моделировании искусственного интеллекта?
3. Дайте определение фактуальному знанию.
4. Дайте определение операционному знанию?
5. Опишите структуру системы, основанной на обработке баз данных.
6. Приведите структуру системы, основанной на обработке знаний.
7. Перечислите признаки ИИС.
8. Перечислите классы ИИС
9. Чем отличаются интеллектуальные базы знаний от традиционных?
10. Для решения каких задач используется естественно-языковой интерфейс?
11. Что является основным элементом экспертной системы?
12. Какие существуют требования к адаптивности ИС?
13. Что такое морфологический анализ?
14. Что такое синтаксический анализ?
15. Что такое семантический анализ?
16. Для чего используется естественно-языковой интерфейс?
17. Для чего предназначены гипертекстовые системы?
18. Что такое системы когнитивной графики?
19. Что такое данные и знания, в чем их отличие?

20. Перечислите этапы трансформации данных.
21. Перечислите этапы трансформации знаний.
22. Дайте определение интенционала и экстенционала понятия.
23. Что такое поверхностные знания, глубинные знания?
24. Перечислите основные модели представления знаний.
25. Что такое продукционная модель?
26. Что такое семантическая сеть?
27. Какие три основных типа отношений используются в семантических сетях?
28. Какие бывают семантические сети?
29. Какие отношения могут использоваться в семантических сетях?
30. Дайте определение фрейма.
31. Перечислите виды фреймов.
32. Как можно представить фрейм через список свойств?
33. Как можно представить фрейм в виде таблицы?
34. Дайте понятие формальной логической теории.
35. Перечислите основные компоненты системы продукций.
36. Что такое рабочая память?
37. Что такое продукционное правило?
38. Какова функция системы управления?
39. Опишите работу основного алгоритма системы продукций.
40. Что такое конфликтное множество?
41. В чем состоит проблема представления для систем продукций?
42. Опишите работу прямой и обратной системы продукций.
43. Дайте определение поля знаний.
44. На какой стадии разработки ЭС формируется поле знаний?
45. Что включает в себя семиотика.
46. Приведите синтаксическую структуру поля знаний.
47. В чем состоит процесс формирования поля знаний?
48. В чем состоит процесс извлечения знаний?

49. В чем состоит процесс приобретения знаний?
50. В чем состоит процесс формирования знаний (machine learning)?
51. Каковы три основных аспекта извлечения знаний?
52. Какой из этих аспектов является ведущим?
53. В чем состоит проблема «общего кода»?
54. В чем состоит процесс формирования понятийной структуры?
55. Какой вид имеет гносеологическая цепочка?
56. В чем суть коммуникативных методов извлечения знаний?
57. В чем суть текстологических методов извлечения знаний?
58. Как можно классифицировать предметные области по степени документированности?
59. Как можно классифицировать предметные области по степени структурированности?
60. Что означает, что предметная область хорошо структурирована?
61. Опишите простейший метод структурирования.
62. Что такое феномен восприятия?
63. Приведите постановку задачи классификации.
64. Как можно охарактеризовать способность к обучению?
65. Приведите математическую постановку задачи обучения.
66. При решении каких задач применяются методы классификации?
67. Какие методы используются для решения задачи классификации?
68. Как строятся дискриминантные функции?
69. Перечислите области применений деревьев решений.
70. Какими достоинствами и недостатками обладают деревья решений?
71. При решении каких практических задач могут применяться методы деревьев решений?
72. Опишите структуру деревьев решений.

6 семестр

1. Дайте определение экспертных систем.
2. Что является основным компонентом экспертной системы?
3. В каких областях применяются экспертные системы?
4. Приведите обобщенную структуру экспертной системы.
5. Опишите работу экспертной системы.
6. Для чего предназначена подсистема объяснений?
7. Для чего предназначен интеллектуальный редактор базы знаний?
8. Что такое база знаний?
9. Приведите классификацию экспертных систем по задаче. Охарактеризуйте каждую из задач.
10. Приведите классификацию экспертных систем по связям с реальным временем.
11. Приведите классификацию экспертных систем по типу ЭВМ.
12. Приведите классификацию экспертных систем по степени интеграции.
13. Перечислите участников коллектива разработчиков экспертной системы.
14. Какие требования предъявляются к пользователю?
15. Каковы функции эксперта в процессе разработки ЭС?
16. Какими качествами должен обладать инженер по знаниям?
17. С какими формами знаний имеет дело инженер по знаниям в процессе разработки ЭС?
18. Перечислите этапы процесса разработки ЭС.
19. Какие задачи решают разработчики на этапе выбора подходящей проблемы?
20. Что такое прототипная система?
21. Перечислите стадии разработки прототипа и охарактеризуйте каждый из них.
22. Опишите процесс перехода от прототипной системы к промышленной ЭС.

23. На основании каких критериев можно проводить оценку системы?
24. В чем суть иерархического подхода при проектировании сложных систем?
25. Как происходит движение между уровнями иерархии в нисходящей концепции?
26. Как происходит движение между уровнями иерархии в восходящей концепции?
27. Перечислите классы подходов к проектированию сложных систем.
28. Опишите объектно-структурный подход к формированию поля знаний.
29. Дайте понятие обучающей выборки.
30. Дайте понятие обучающей выборки «с учителем».
31. Дайте понятие обучающей выборки «без учителя».
32. Как осуществляется процесс классификации примеров обучающей выборки в индуктивных системах?
33. Перечислите основные проблемы, решаемые нейронными сетями.
34. Что такое полносвязная нейронная сеть?
35. Что такое слабосвязная нейронная сеть?
36. Что такое многослойная нейронная сеть?
37. Что такое нейронная сеть прямого распространения?
38. Приведите математическую модель нейрона.
39. Перечислите виды функций активации нейрона.
40. Приведите архитектуру многослойной сети прямого распространения.
41. Запишите формулы, в соответствии с которыми происходит функционирование многослойной сети прямого распространения.
42. Нарисуйте график пороговой функции активации.
43. Нарисуйте график экспоненциальной сигмоиды.
44. Что такое обучающий пример?
45. Приведите примеры обучающих примеров для различных практи-

ческих задач.

46. Что значит обучить многослойную сеть прямого распространения?
47. В чем суть алгоритма обратного распространения ошибки?
48. Чем отличаются обучение на всем множестве примеров, одиночное предъявление примеров и постраничное обучение?
49. Приведите формулировку теоремы о полноте.

Критерии оценивания компетенций

Оценка «**зачтено**» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения. Допускаются некоторые неточности, недостаточно правильные формулировки в изложении программного материала, затруднения при выполнении практических работ.

Оценка «**не зачтено**» выставляется студенту, если он не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические задания.

3. ОТЧЕТНОСТЬ ПО ДИСЦИПЛИНЕ

Успеваемость студентов по дисциплине оценивается в ходе текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация

Процедура зачета как отдельное контрольное мероприятие не проводится, оценивание знаний обучающегося происходит по результатам текущего контроля.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Перечень основной литературы:

1. Исаев, С. В. Интеллектуальные системы : учебное пособие / С. В. Исаев, О. С. Исаева. — Красноярск : Сибирский федеральный университет, 2017. — 120 с. — ISBN 978-5-7638-3781-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/84365.html> (дата обращения: 16.01.2024). — Режим доступа: для авторизир. пользователей.
2. Представление знаний в информационных системах : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, М. Ю. Серегин [и др.]. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2012. — 169 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/64163.html> (дата обращения: 16.01.2024). — Режим доступа: для авторизир. пользователей.

Перечень дополнительной литературы:

1. Загорулько, Ю. А. Инженерия знаний : учебное пособие / Ю. А. Загорулько, Г. Б. Загорулько. — Новосибирск : Новосибирский государственный университет, 2016. — 93 с. — ISBN 978-5-4437-0452-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/93454.html> (дата обращения: 24.07.2023). — Режим доступа: для авторизир. пользователей.
2. Цильковский, И. А. Методы анализа знаний и данных : конспект лекций / И. А. Цильковский, В. М. Волкова. — Новосибирск : Новосибирский государственный технический университет, 2010. — 68 с. — ISBN 978-57782-1377-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/45385.html> (дата обращения: 16.01.2024). — Режим доступа: для авторизир. пользователей.